Dokumentacja aplikacji

System pomocy bezdomnym zwierzętom

Wykonali:

1. Mateusz Głód
2. Piotr Tyburczy
3. Emilia Zamorowska

1. Opis aplikacji

1.1. Główne założenia

Aplikacja Doggo. skupia się na funkcjonalnościach związanych z ogólno pojętą pomocą zwierzętom. Swoją działalność opiera na wspieraniu schronisk w odnalezieniu bezdomnym zwierzętom domu, jak i na ratowaniu posiadających właścicieli zaginionych pupili. Aplikacja webowa umożliwia zalogowanym użytkownikom zgłoszenie zaginięcia. Można dodawać je w łatwy sposób za pomocą prostego formularza oraz mapy. Wszyscy mają dostęp do mapy zaginięć. To na niej pojawiają się pinezki, do których przypięte są informacje na temat zwierząt, które aktualnie zaginęły. Schroniska dodatkowo mogą skorzystać z możliwości zgłoszenia zwierzęcia do adopcji i udostępnienia swoich zapotrzebowań na różnego typu towary, które występują w danym schronisku. Aplikacja wyświetla informacje użytkownikom o zapotrzebowaniach schronisk oraz umożliwia zgłoszenie swojego zainteresowania adopcją. Każdy zalogowany użytkownik posiada możliwość przejrzenia i usunięcia zgłoszeń oraz adopcji (w przypadku schroniska) dodanych przez swoje konto.

Niezalogowany użytkownik będzie mógł korzystać tylko z wyświetlania informacji w aplikacji. Aby dodawać zgłoszenia potrzebne jest konto użytkownika. Ma ono przypisaną rolę, nadaną przy rejestracji. Istnieją dwa typy profili: użytkownik prywatny (osoba prywatna) oraz schronisko.

Wymagania funkcjonalne

- **Użytkownik niezalogowany**
    - Logowanie
    - Rejestracja
    - Dostęp do listy zaginięć
    - Dostęp do listy aktualnych adopcji
    - Dostęp do listy profilów oraz zapotrzebowań schronisk

- **Użytkownik zalogowany (dodatkowo)**
    - Zgłaszanie zaginięć
    - Odnawianie zaginięcia
    - Umieszczanie ogłoszeń o adopcji (schronisko)
    - Usuwanie adopcji (schronisko)
    - Dostęp do listy adopcji dodanych przez określonego użytkownika (schronisko)
    - Dostęp do listy zaginięć dodanych przez określonego użytkownika

Wymagania niefunkcjonalne

- Aplikacja zgodna z najpopularniejszymi przeglądarkami
- Udostępnienie graficznego interfejsu przyjaznego użytkownikom
- Zostaną zachowane wszystkie wymagane zasady bezpieczeństwa zarządzania danymi użytkowników

1.2. Cele aplikacji

Głównym założeniem aplikacji była łatwość w korzystaniu z niej. Każdy użytkownik wykorzystujący nasz system ma być w stanie szybko oznaczyć zwierzę jako zaginione. Dzięki temu inne osoby przeglądające stronę niezwłocznie mogą mieć dostęp do dokładnej ostatniej lokalizacji zwierzęcia, co usprawnia poszukiwania. Celem aplikacji było również wspieranie lokalnych schronisk, które mogą udostępniać na bieżąco swoje aktualne zapotrzebowania, a użytkownicy w ramach wolontariatu, je uzupełniać. Kolejnym celem było stworzenie aplikacji, która pomogłaby w różnego typu adopcjach zwierząt. Użytkownicy prywatni mają dostęp do oznaczania chęci adopcji danego zwierzęcia, a schroniska takie adopcje mogą tworzyć.

1.3. Technologie wykorzystane w projekcie

Frontendowe	Backendowe	Bazodanowe	Komunikacja
HTML5	Node.js	Mongo.db	REST API
CSS3			
Bootstrap			
Typescript			
React.js			

Frontend

HTML5

Jest to język znaczników wykorzystywany do tworzenia i prezentowania stron internetowych. Jest rozwinięciem języka HTML 4 i jego XML-owej odmiany (XHTML 1), opracowywanej w ramach prac grupy roboczej WHATWG (Web Hypertext Application Technology Working Group) i W3C.

CSS3

Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. Arkusz stylów CSS to lista dyrektyw (tzw. reguł) ustalających w jaki sposób ma zostać wyświetlana przez przeglądarkę internetową zawartość wybranego elementu (lub elementów) (X)HTML lub XML

Bootstrap

Biblioteka CSS, rozwijana przez programistów Twittera, wydawany na licencji MIT. Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie interfejsu graficznego stron oraz aplikacji internetowych. Bazuje głównie na gotowych rozwiązaniach HTML oraz CSS (kompilowanych z plików Less) i może być stosowany m.in. do stylizacji

takich elementów jak teksty, formularze, przyciski, wykresy, nawigacje i innych komponentów wyświetlanych na stronie. Biblioteka korzysta także z języka JavaScript.

TypeScript

TypeScript – wolny i otwartoźródłowy język programowania stworzony przez firmę Microsoft jako nadzbiór języka JavaScript. Umożliwia on opcjonalne statyczne typowanie oraz programowanie zorientowane obiektowo oparte na klasach. TypeScript jest nadzbiorem JavaScript, a więc potencjalnie każdy program napisany w języku JavaScript jest poprawnym programem TypeScript. Aplikacje napisane w TypeScript kompilują się bezpośrednio do języka JavaScript.

React.js

React.js (inne stosowane nazwy: React, ReactJS) – biblioteka języka programowania JavaScript, która wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych. Została stworzona przez Jordana Walke, programistę Facebooka, a zainspirowana przez rozszerzenie języka PHP – XHP. Często wykorzystywana do tworzenia aplikacji typu Single Page Application.

Backend

Node.js

Node.js – wieloplatformowe środowisko uruchomieniowe o otwartym kodzie[3] do tworzenia aplikacji typu server-side napisanych w języku JavaScript. Przyczynił się do stworzenia paradygmatu „JavaScript everywhere” umożliwiając programistom tworzenie aplikacji w obrębie jednego języka programowania zamiast polegania na odrębnych po stronie serwerowej.

Bazodanowe

Mongo.db

MongoDB – otwarty, nierelacyjny system zarządzania bazą danych napisany w języku C++. Charakteryzuje się brakiem ściśle zdefiniowanej struktury obsługiwanych baz danych. Zamiast tego dane składowane są jako dokumenty w stylu JSON.

Komunikacja

REST API

Representational state transfer, REST – styl architektury oprogramowania wywiedziony z doświadczeń przy pisaniu specyfikacji protokołu HTTP dla systemów rozproszonych. REST wykorzystuje m.in. jednorodny interfejs, bezstanową komunikację, zasoby, reprezentacje, hipermedia, HATEOAS.

Zewnętrzne API Imgur

W celu przechowywania zdjęć na stronie wykorzystane zostało API serwisu Imgur. Umieszczane zdjęcia zapisywane są w bazie danych Imgur. Baza wykorzystana w projekcie przechowuje jedynie linki do zdjęć. Dzięki temu w bazie mongodb zapisywana jest mniejsza ilość danych.

2. Diagramy

2.1. Diagram przypadków użycia

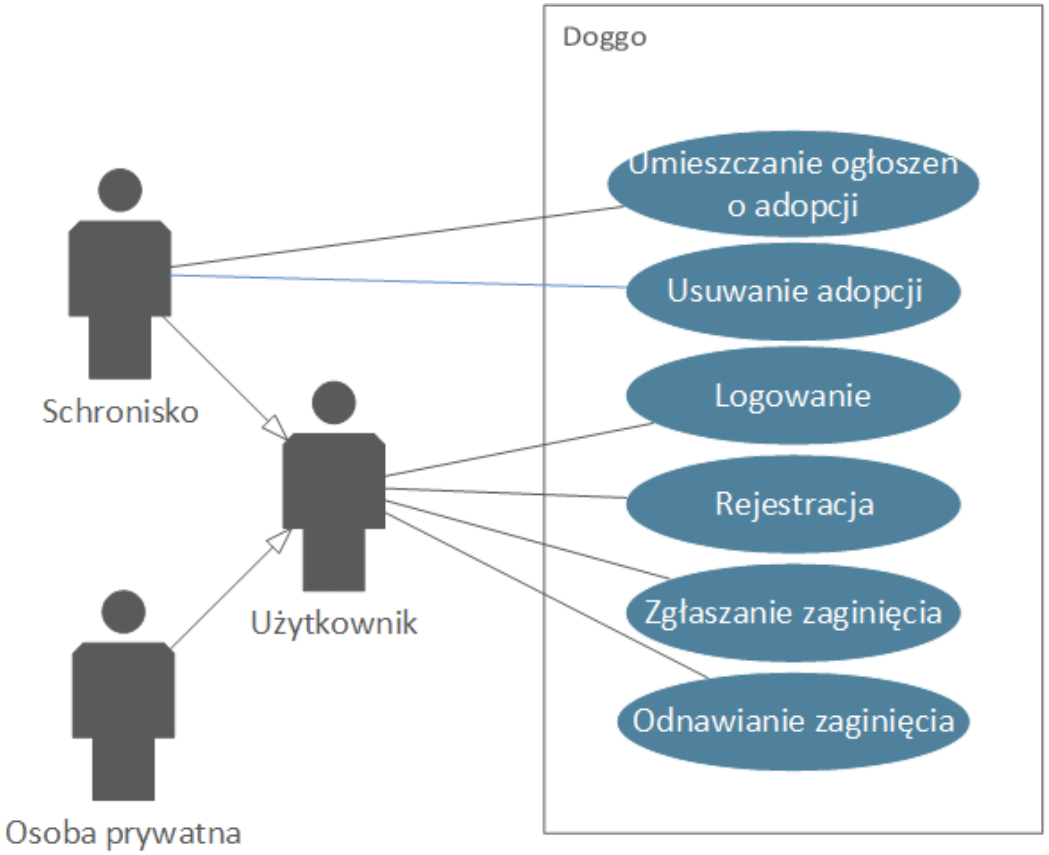


Diagram przedstawia podstawowe funkcjonalności systemu Doggo wraz z ich otoczeniem. Widoczny jest tutaj podział użytkowników ze względu na przypisane im role.

2.2. Diagramy sekwencji

Zgłoszenie zaginięcia

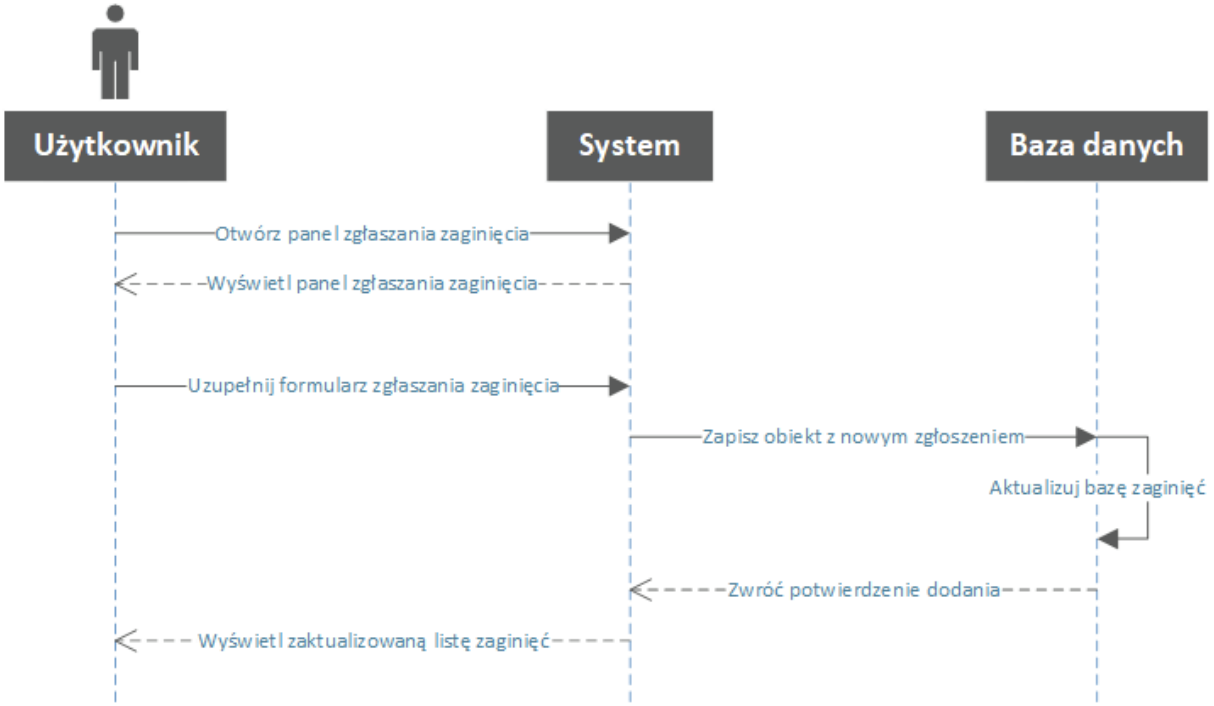


Diagram prezentujący kolejne sekwencje operacji zgłaszania zaginięcia zwierzęcia. Użytkownik w celu zgłoszenia wypełnia formularz zgłoszeniowy. System zapisuje w bazie nowe zaginięcie.

Zgłoszenie zwierzęcia do adopcji

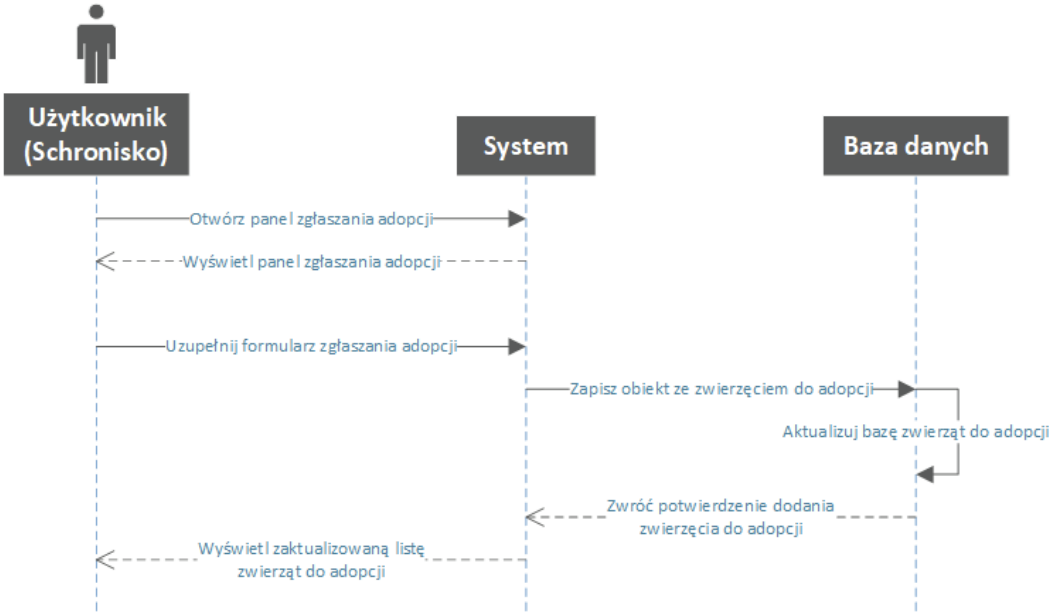


Diagram prezentujący kolejne sekwencje operacji zgłaszania zwierzęcia do adopcji. Użytkownik - schronisko w celu zgłoszenia adopcji wypełnia formularz zgłoszeniowy. System zapisuje w bazie nowe zwierzę do adopcji.

Usuwanie adopcji

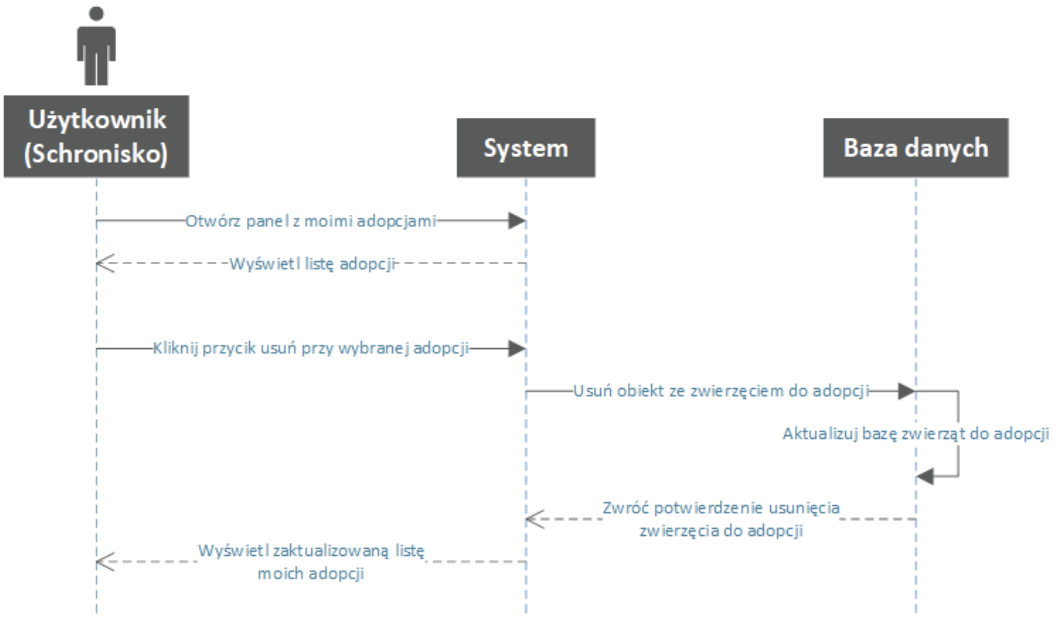


Diagram prezentujący kolejne sekwencje operacji usuwania ogłoszenia o adopcję. Użytkownik - schronisko w celu usunięcia adopcji wciska przycisk przy wybranej

pozycji. System usuwa z bazy wybraną adopcję.

Odnawianie przedawnionego zaginięcia

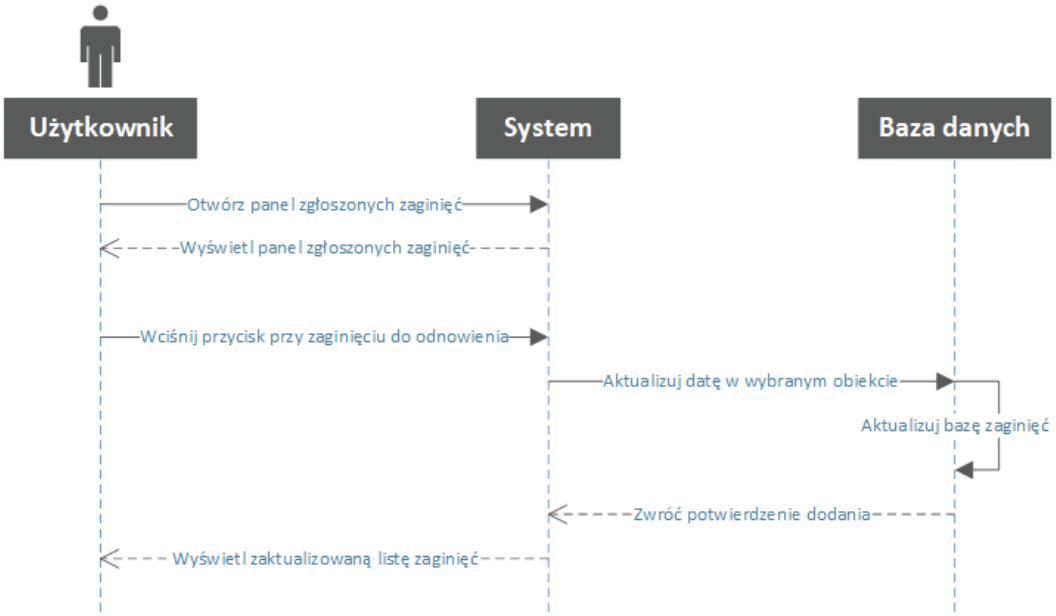


Diagram prezentujący kolejne sekwencje operacji odnawiania przedawnionego zaginięcia. Użytkownik w celu odnowienia zaginięcia wciska przycisk przy wybranej pozycji. System zmienia datę zgłoszenia w bazie danych.

3. Prezentacja aplikacji

3.1. Opis bazy danych

Modele danych przechowywane są w bazie MongoDB. Stworzono 4 schematy niezbędne do magazynowania informacji w aplikacji.

- **Adoption**

  Zawiera odpowiednie pola odpowiadające między innymi za połączenie użytkownika z adopcją oraz adopcji z dodającym ją schroniskiem. Prawie wszystkie właściwości są oznaczone jako wymagane. Pole *userMail* jest uzupełniane dopiero, gdy któryś z użytkowników zgłosi chęć adopcji danego zwierzęcia. Zdjęcia przechowywane są tutaj za pomocą linków, automatycznie generowanych po wrzuceniu przez użytkownika pliku ze zdjęciem.

```
const adoptionSchema = mongoose.Schema({
    id :String,
    name: { type: String, required: true },
    userMail :String,
    shelterName :{ type: String, required: true },
    image: { type: String, required: true },
    description: { type: String, required: true },
});
```

- **Missing**

  Przechowuje obiekt zawierający informacje niezbędne do wyświetlania zaginięcia w formie opisu, jak i za pomocą znacznika na mapie (właściwości *longitude* oraz *latitude*). Tutaj także zdjęcia przechowywane są za pomocą linków.

```
const missingSchema = mongoose.Schema({
  id: String,
  title: { type: String, required:  true },
  image: { type: String, required:  true },
  creator: { type: String, required:  true },
  time: { type: Date, required:  true },
  description: { type: String, required:  true },
  place: { type: String, required:  true },
  longitude:  { type: Number, required:  true },
  latitude:  { type: Number, required:  true },
});
```

- **Shelter**

Jest to model przechowujący konto użytkownika z przypisaną rolą schroniska.

```
const shelterSchema = mongoose.Schema({
  name: { type: String, required:  true },
  email: { type: String, required: true },
  password: { type: String, required: true },
  id: { type: String },
  NIP: { type: String, required: true },
  food: { type: String },
  equipment: { type: String },
});
```

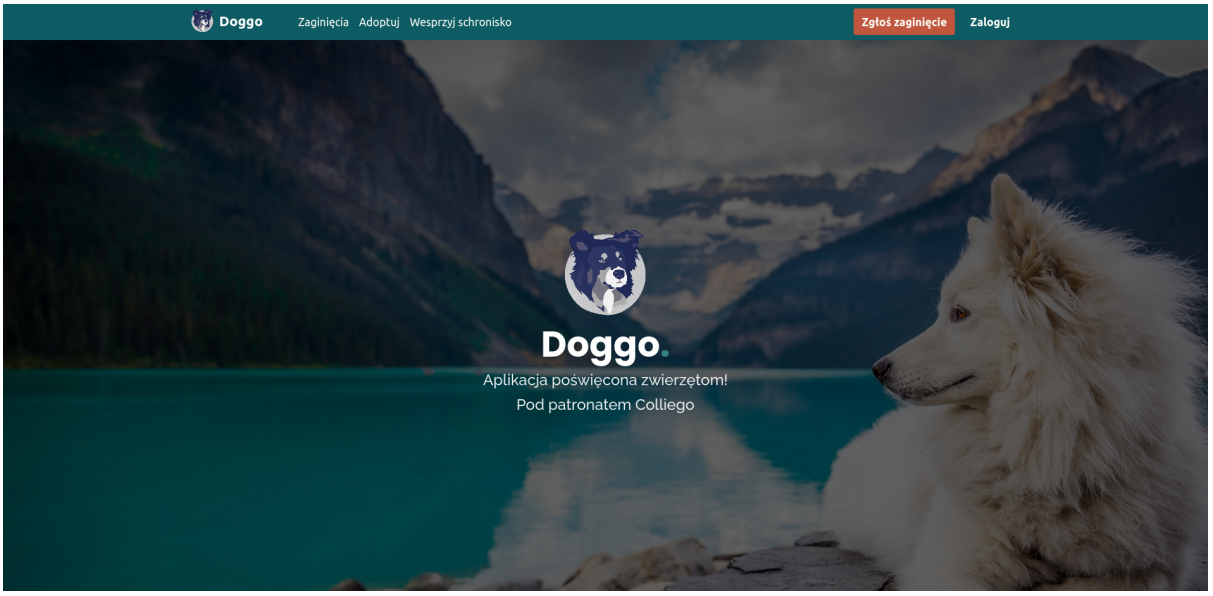
- **User**

Model przechowujący konto użytkownika z przypisaną rolą osoby prywatnej

```
const userSchema = mongoose.Schema({
  name: { type: String, required:  true },
  email: { type: String, required: true },
  password: { type: String, required: true },
  id: { type: String }, });
```

3.2. Główne funkcjonalności oraz widoki aplikacji

Strona główna



Rys. 3.1. Strona główna aplikacji

Strona główna nie zawiera wiele funkcjonalności, jest czysto reprezentatywna. Odpalając aplikację ukazuje się nam podstawowe logo oraz tło strony. Na górze znajduje się pasek nawigacyjny, a poniżej opisane są różne elementy aplikacji.

Rys. 3.2. Strona główna - założenia aplikacji

Przewijając stronę w dół, ukazują się różne opisy i założenia jakie przyświecały nam podczas tworzenia aplikacji (Rys. 3.2). Przedstawienie założeń aplikacji jest nie tylko

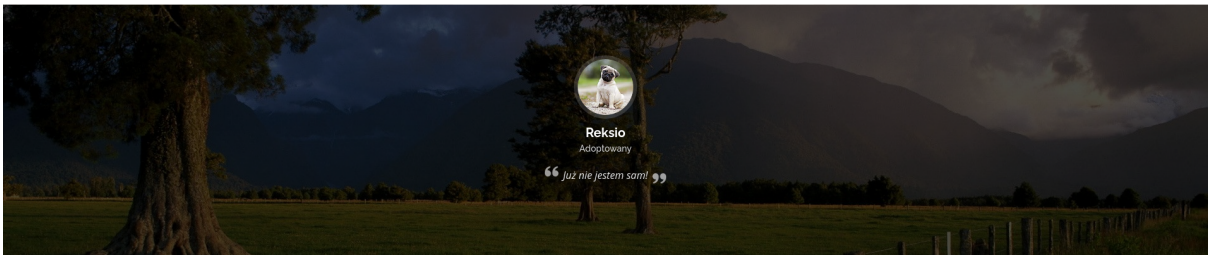
dobre dla twórców, ale i uświadamia potencjalnych użytkowników ile dobrego mogą oni uczynić korzystając z tak prostej strony.

Rys. 3.3. Strona główna - ciąg dalszy

W tym miejscu (Rys.3.3) znajdują się podstawowe informacje na temat aplikacji oraz jej funkcjonalności, zostało to opisane pokrótce, jedynie po to by zachęcić użytkowników do korzystania z niej.

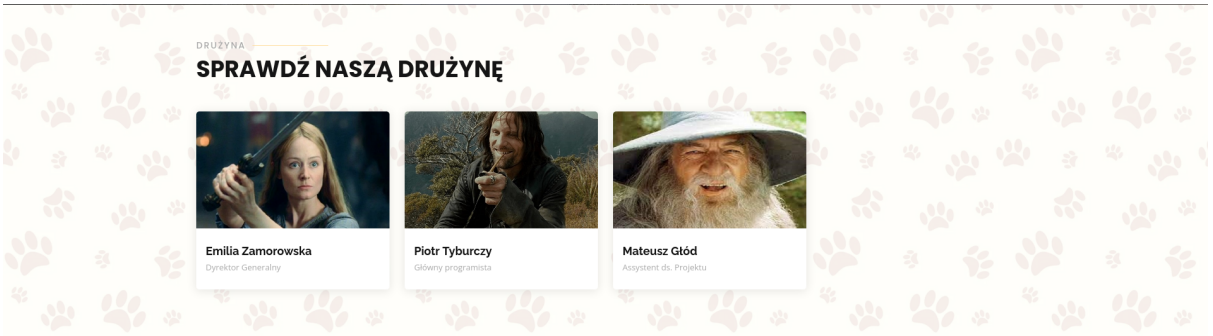
Rys. 3.4. Strona główna - ciąg dalszy

W kolejnej sekcji (Rys. 3.4) są pokazane liczniki, dzięki którym użytkownicy mogą zobaczyć co do tej pory osiągnęła aplikacja.



Rys. 3.5. Strona główna - ciąg dalszy

W następnej sekcji (Rys. 3.5) przedstawiony jest dany pupil, który do tej pory został adoptowany. Sekcja zwraca ponownie uwagę użytkowników na potencjalną możliwość adopcji zwierząt.



Rys. 3.6. Strona główna - twórcy

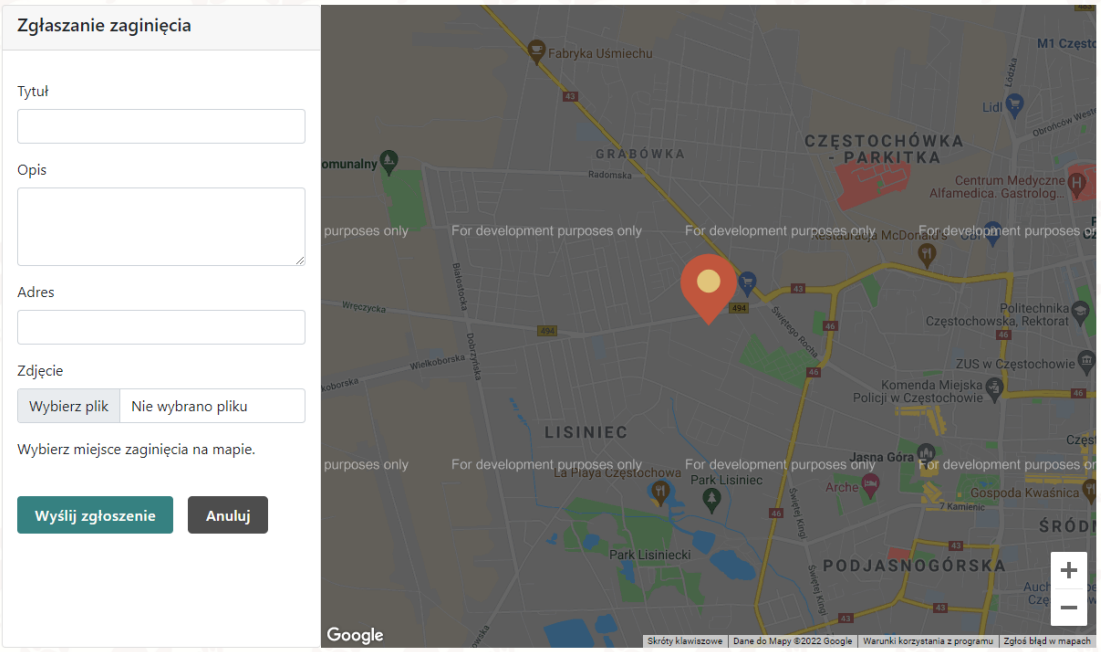
Kolejna sekcja jest poświęcona twórcom (Rys. 3.6). Praca nad takim projektem nie powinna być powodem do wstydu, dlatego też zostali oni jawnie przedstawieni na stronie głównej aplikacji.

Rys. 3.7. Strona główna - współprace

Ostatnia część strony głównej przedstawia aktualnie współpracujące z zespołem jednostki. Projekt w założeniu powinien być ogólnodostępny i powinien nawiązywać współpracę z coraz to liczniejszymi schroniskami czy innymi partnerami.

Zgłoszenie zaginięcia

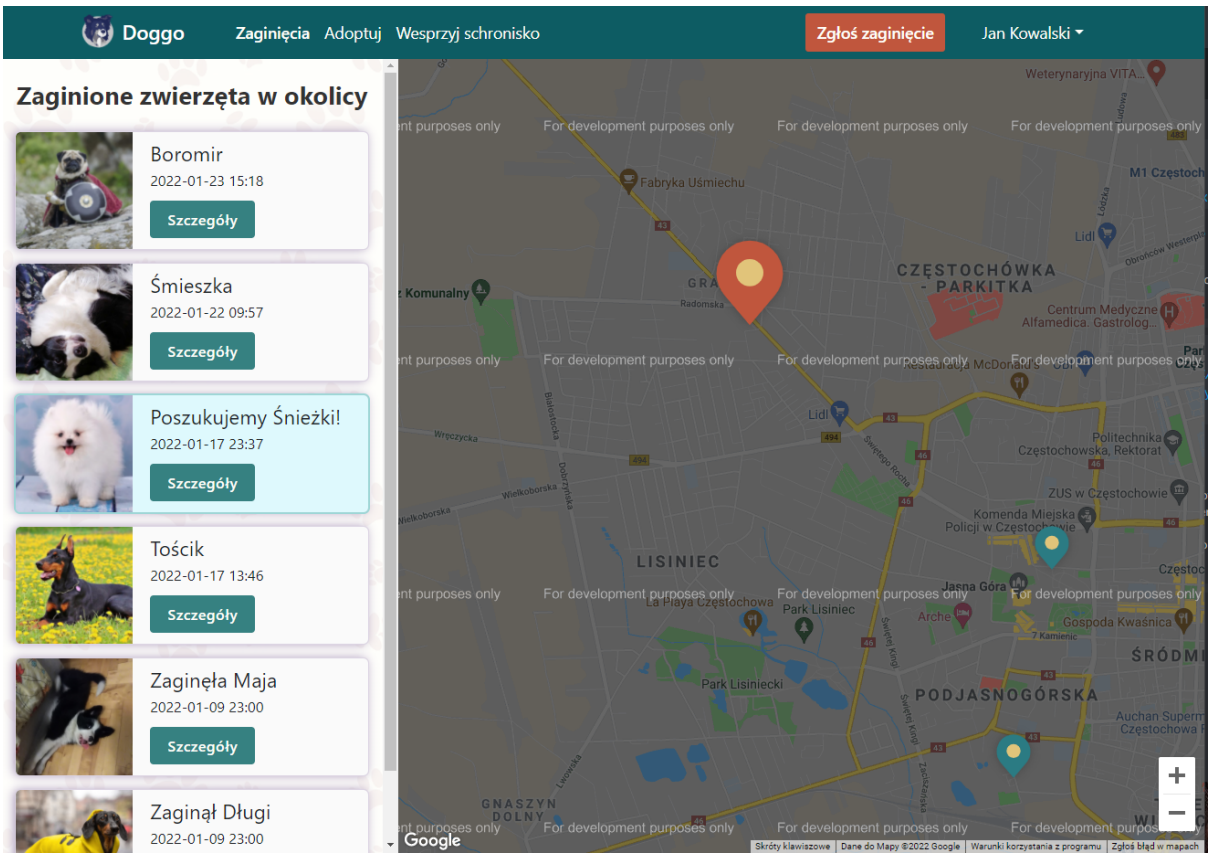
Do formularza zgłaszania zaginięcia odsyła przycisk *Zgłoś zaginięcie* widoczny na pasku nawigacyjnym strony.



Rys. 3.8. Formularz dodawania zgłoszenia o zaginięciu

Zgłoszenie zaginięcia jest dostępne dla wszystkich zalogowanych użytkowników aplikacji. Formularz dodawania zawiera odpowiednie pola potrzebne do zapisania go w bazie danych oraz mapę, na której użytkownik wybiera miejsce, gdzie ostatnio był widziany jego pupil. Tożsamość dodającego oraz data pobierane są automatycznie przez aplikację. Jeśli dane zadeklarowane przez użytkownika są uzupełnione poprawnie, baza aktualizuje kolekcję zaginięć, a użytkownik zostaje przekierowany na stronę zawierającą wszystkie zgłoszenia.

Wyświetlanie zaginięć

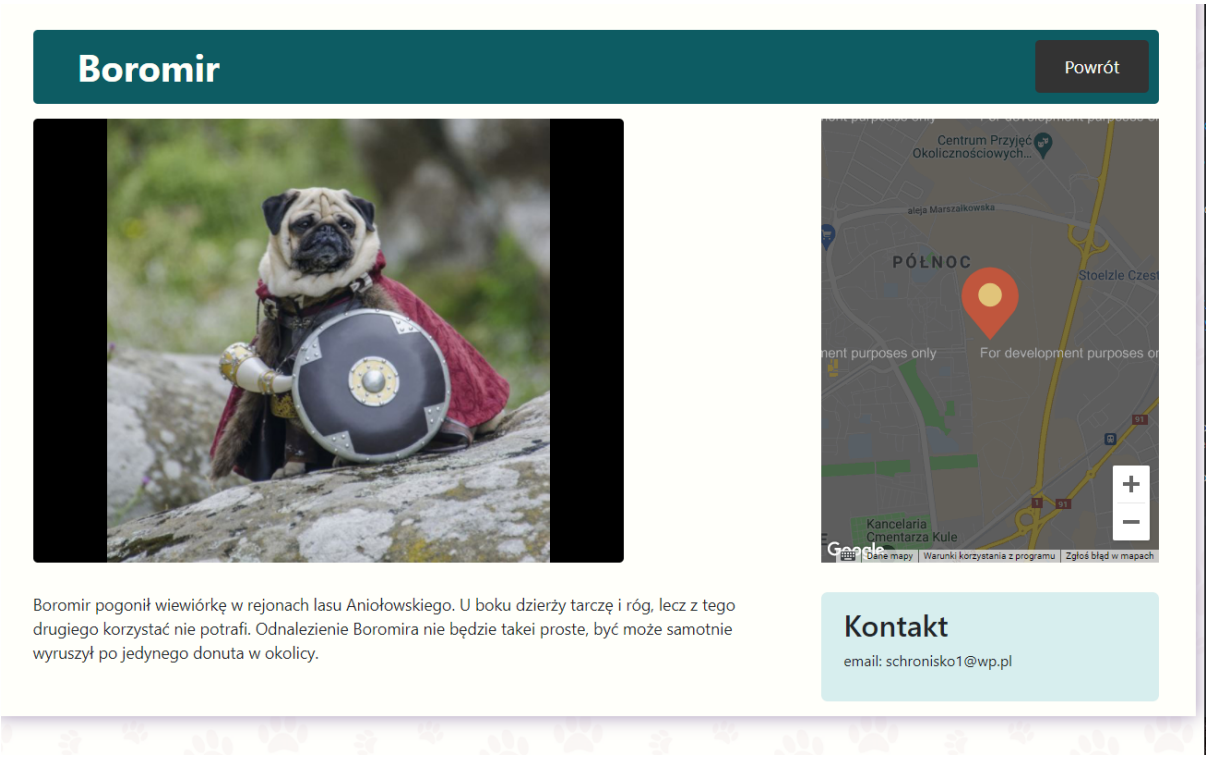


Rys. 3.9. Widok zaginięć

Widok podstrony *Zaginięcia* podzielony jest na dwie części. Po lewej stronie wyświetlana jest lista niedawno dodanych zgłoszeń. Jest ona posortowana po dacie dodania, od najnowszego. Aby zapobiec przepełnieniu listy, zgłoszenia zyskują status przedawnionego po 30 dniach od dodania i przestają być na niej wyświetlane. Nie zostają one jednak usuwane z bazy - użytkownik nadal będzie widział dodane przez siebie przedawnione zgłoszenie na swoim profilu. Będzie mógł on w dowolnym momencie usunąć je lub dodać ponownie.

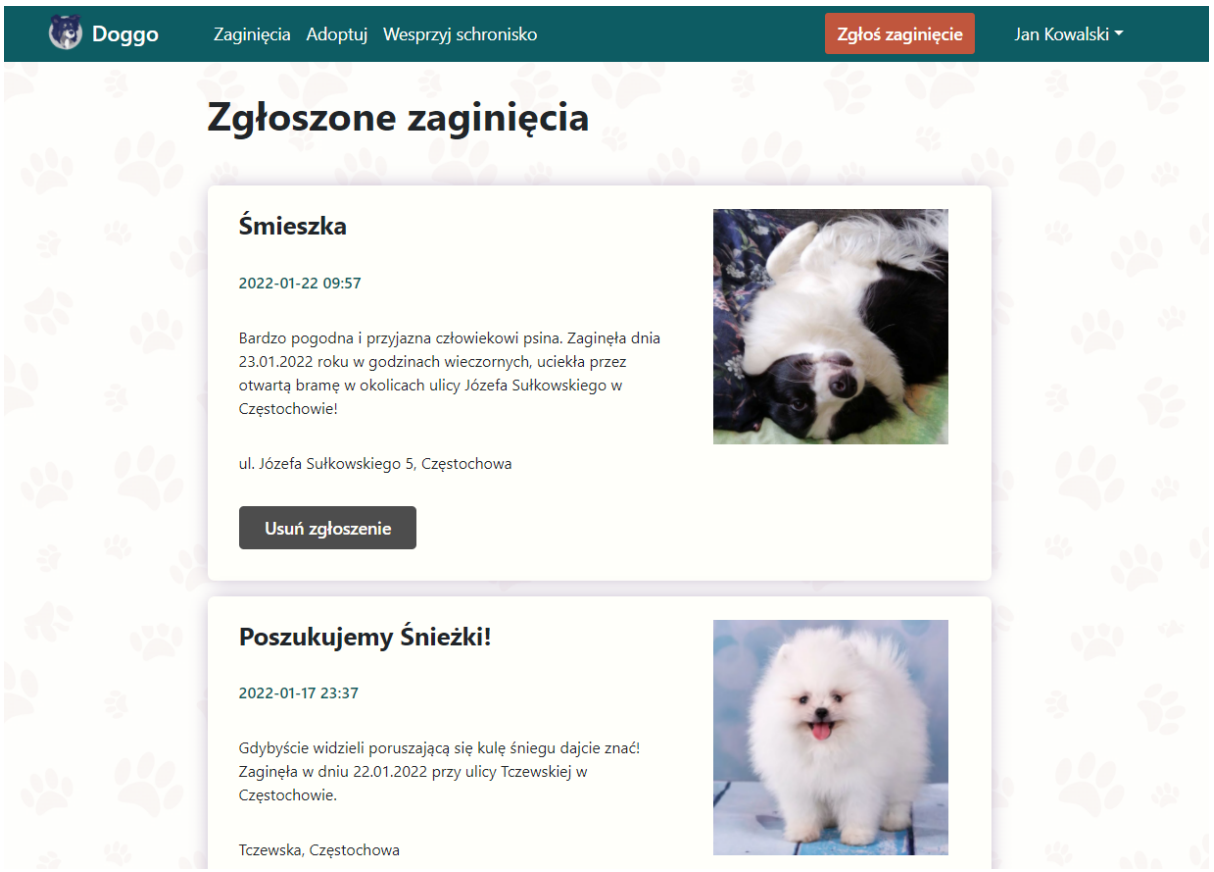
Po prawej części strony wyświetlana jest mapa ze znacznikami wskazującymi na lokalizację każdego widocznego na liście zgłoszenia. Mapa pobiera lokalizację urządzenia i dokonuje wyśrodkowania widoku na dany obszar. Aktualnie zaznaczona pozycja na liście zaginięć jest oznaczona na mapie czerwonym znacznikiem. Kliknięcie na daną pinezkę na mapie zaznacza także odpowiedni element na liście.

Każde zgłoszenie może zostać wyświetlone za pomocą bardziej szczegółowego widoku po kliknięciu przycisku *Szczegóły*.



Rys. 3.10. Widok szczegółowy pojedynczego zaginięcia

Wyświetlanie zaginięć dodanych przez zalogowanego użytkownika

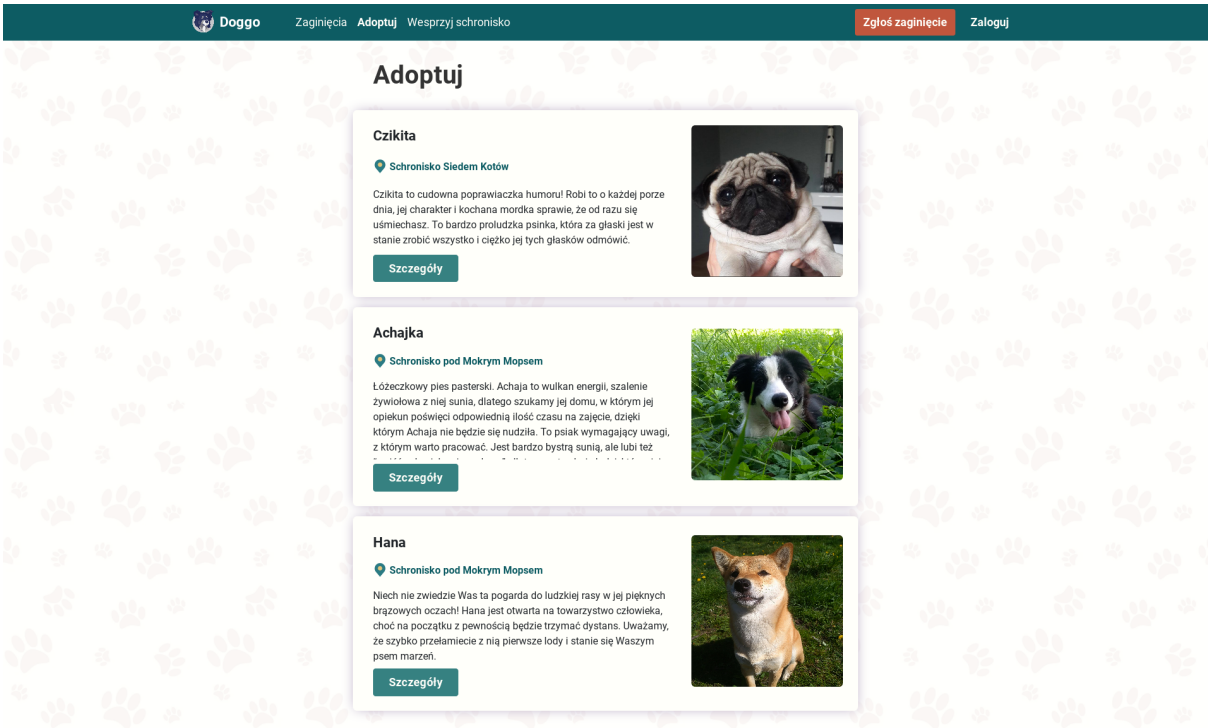


Rys. 3.11. Zakładka *Zgłoszone zaginięcia* danego użytkownika

Dla zalogowanego użytkownika dostępna jest zakładka Zgłoszone zaginięcia. Przy każdym elemencie listy dostępny jest przycisk usunięcia. Zadbano o wyróżnienie zgłoszeń przedawnionych - są one podświetlane kolorem czerwonym zachęcając użytkownika do podejmowania akcji z nimi związanych - ostatecznego usunięcia zaginięcia z bazy lub odnowienia go przypisując mu aktualną datę.

Rys. 3.12. Widok przedawnionego zaginięcia

Wyświetlanie adopcji



Rys. 3.13. Zakładka *Adoptuj*

Do widoku dostępnych adopcji prowadzi przycisk *Adoptuj* na pasku nawigacyjnym. Znajduje się tu lista zwierząt, które zostały zgłoszone do adopcji przez schroniska. Użytkownicy mają dostęp do wyświetlania szczegółów danego zgłoszenia.

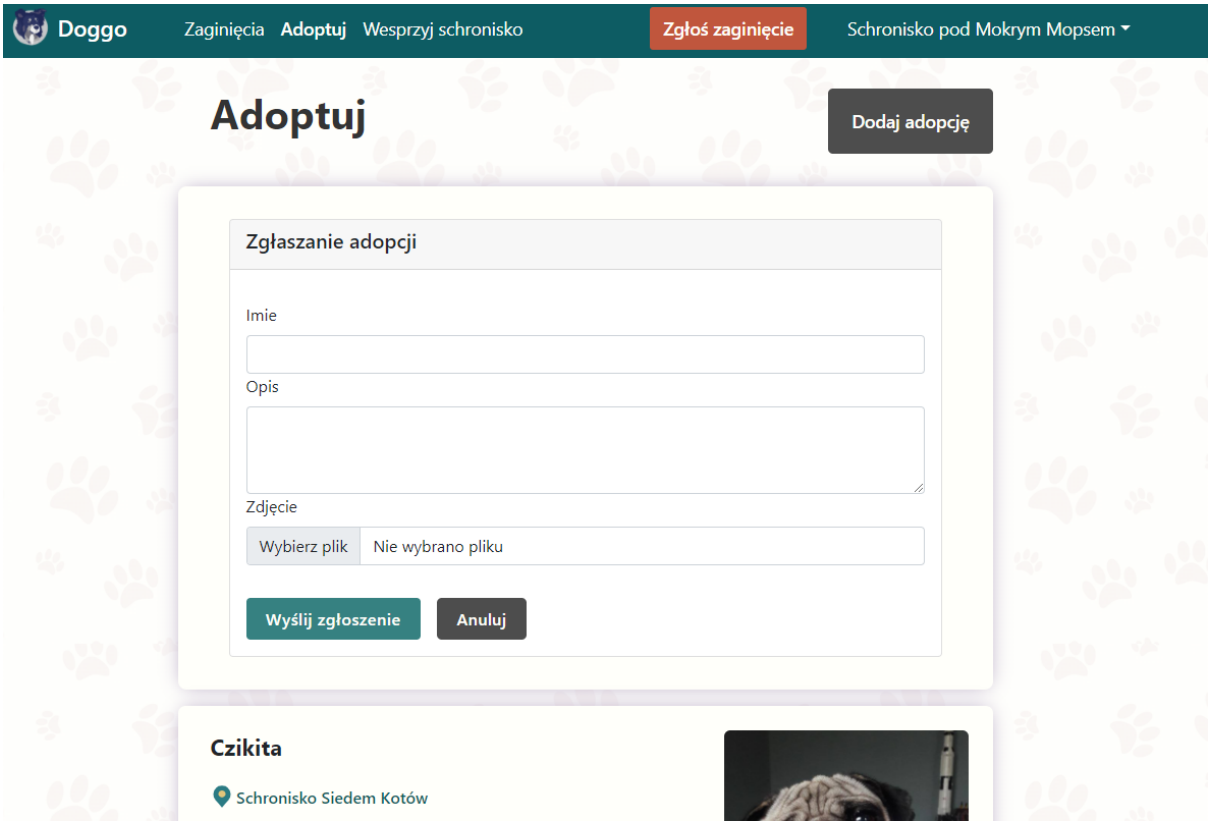
Rys. 3.14. Szczegóły pojedynczej adopcji

Użytkownik po kliknięciu na szczegóły może zgłosić chęć adopcji. Opcja ta jest dostępna tylko dla osoby prywatnej, gdy żaden inny użytkownik nie zrobił tego wcześniej. Jeśli zwierzęciem już ktoś się interesuje, wyświetlony zostaje odpowiedni komunikat.

Dodawanie adopcji

Jeśli zalogowanym użytkownikiem jest schronisko, wyświetlane są dodatkowe opcje dostępne tylko dla użytkowników z tą rolą. Jest to przycisk dodawania adopcji, dzięki któremu zostaje rozwinięty odpowiedni formularz. Dodatkowo przy adopcjach

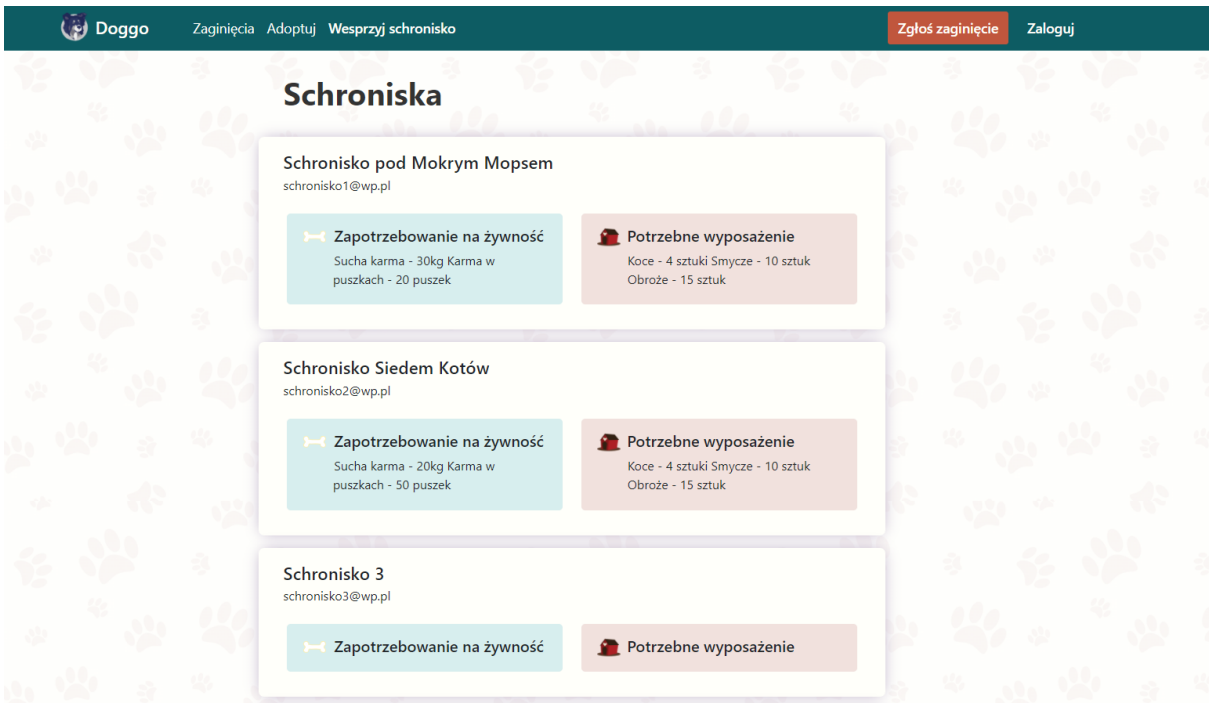
dodanych przez zalogowane schronisko wyświetla się możliwość usunięcia zgłoszenia.



Rys. 3.15. Możliwość dodawania adopcji przez schronisko

Rys. 3.16. Możliwość usunięcia ogłoszenia przez schronisko.

Wyświetlanie listy schronisk

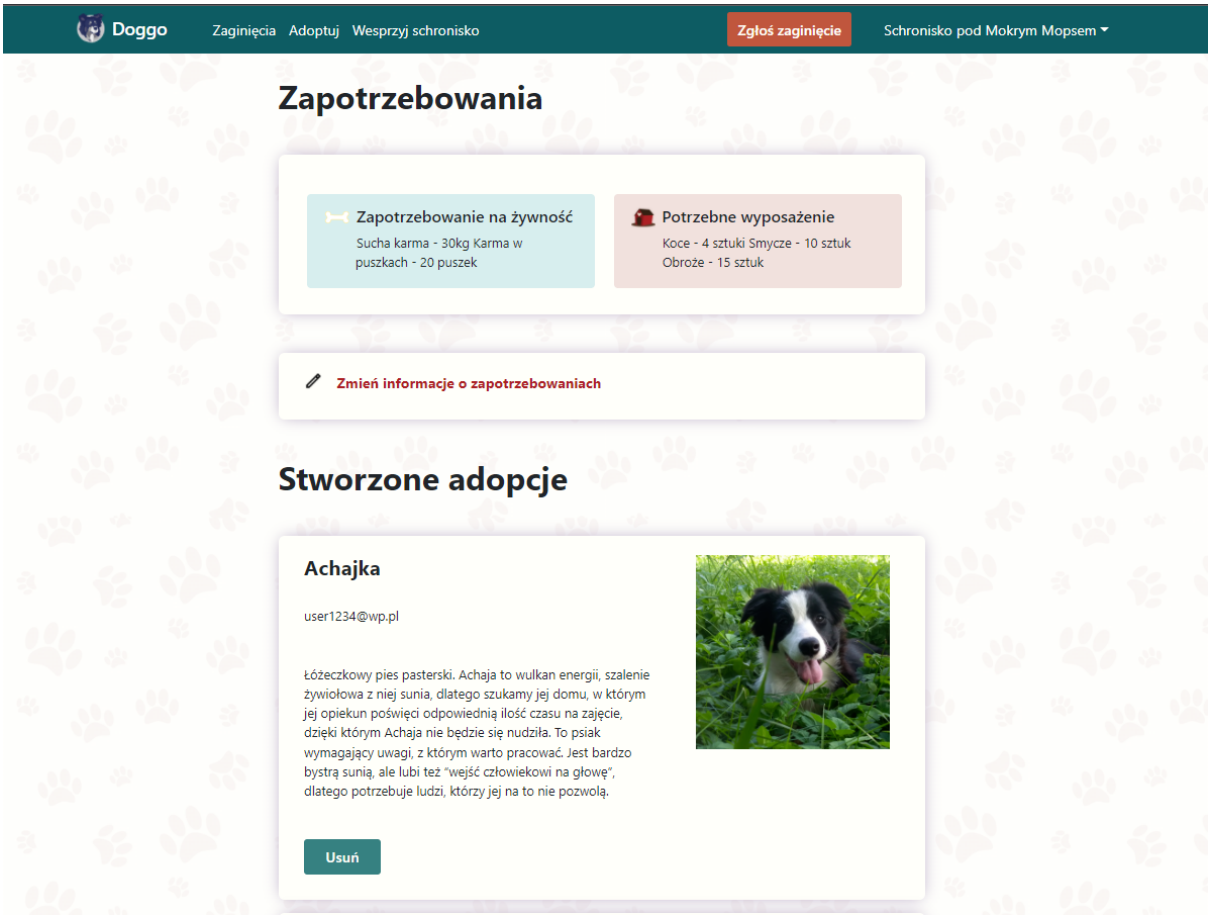


Rys. 3.17. Zakładka *Wesprzyj schronisko*

Stronę poświęconą wsparciu schronisk można znaleźć w zakładce Wesprzyj schronisko. Spisane są tam informacje na temat schronisk, które wystawiły swoje zapotrzebowania na dane materiały. Znajdują się tu ich nazwy, podstawowy kontakt do nich oraz wszelkie zadeklarowane zapotrzebowania.

Zakładka *Moje konto*

Na tej podstronie można znaleźć wszelkie informacje na temat konta danego użytkownika. W przypadku schroniska znajdują się tu zapotrzebowania, które można edytować za pomocą czerwonego przycisku *zmień informacje o zapotrzebowaniach*. Rozwija on odpowiedni formularz z aktualnymi danymi podanymi przez schronisko. Dodatkowo wyświetlają się tutaj wszelkie adopcje, które schronisko stworzyło. W przypadku osoby prywatnej widzimy tutaj adopcje, do których została zgłoszona chęć przystąpienia. W tym miejscu osoba prywatna może zrezygnować z chęci adopcji.

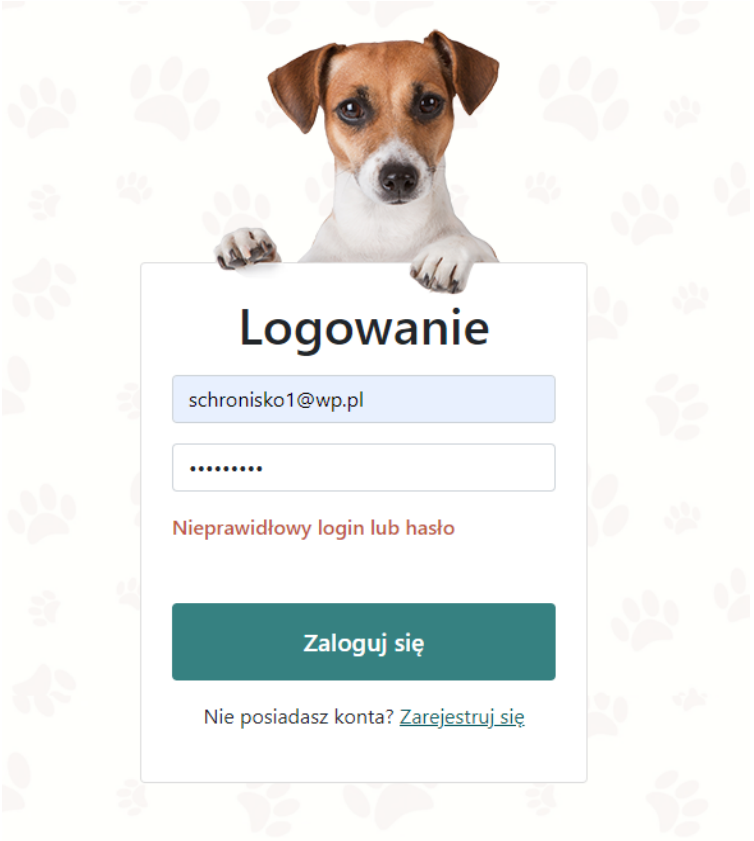


Rys. 3.18. Widok zakładki *Moje konto* dla schroniska

Rys. 3.19. Widok zakładki *Moje konto* dla osoby prywatnej

Logowanie

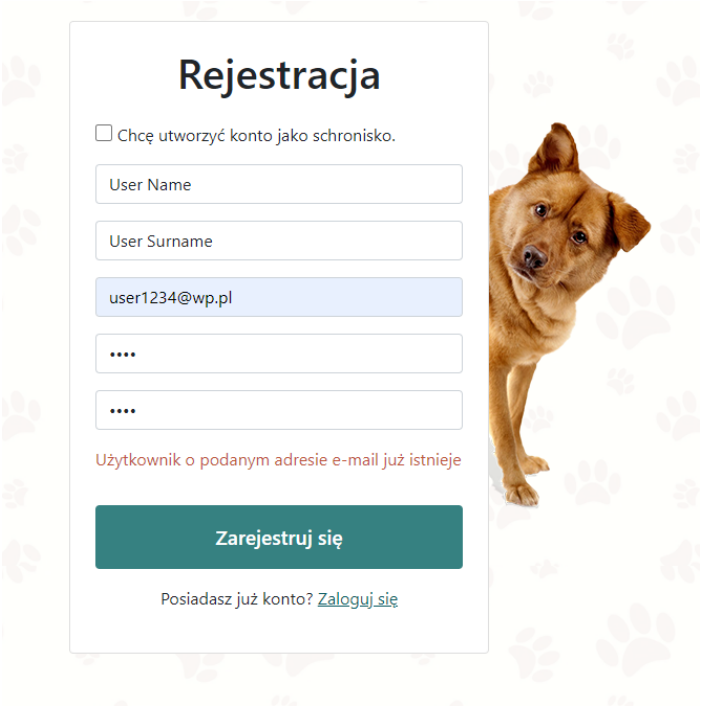
Dane użytkownika przechowywane są w elemencie localStorage przeglądarki. Każde hasło jest odpowiednio haszowane dla zachowania bezpieczeństwa. Po każdym zalogowaniu utworzony jest odpowiedni token, który później wykorzystywany jest do autoryzacji określonych zadań. Wprowadzona została walidacja pól formularza. W razie złych danych logowania, wyświetlony zostaje odpowiedni komunikat.



Rys. 3.20. Okno logowania wraz z rezultatem niepoprawnej próby logowania

Rejestracja

Rejestracja osoby prywatnej i schroniska została rozdzielona w interfejsie użytkownika jak i w samym kodzie aplikacji. Podczas sprawdzania, czy w bazie nie ma już podanego użytkownika brany jest pod uwagę adres e-mail. W przypadku schroniska unikatowość musi występować w ramach pól e-mail, nazwa oraz NIP.



Rys. 3.21. Okno rejestracji osoby prywatnej wraz z informacją o błędzie

Rys. 3.22. Okno rejestracji schroniska wraz z informacją o błędzie

4. Uruchamianie aplikacji

Aplikacja może zostać uruchomiona na dwa sposoby.
Dzięki hostingowi, istnieje możliwość wyświetlenia aplikacji bezpośrednio za pomocą linku: https://grupowy-pies.herokuapp.com/
Jest to preferowana metoda uruchamiania, ponieważ tylko tutaj działa przesyłanie zdjęć podczas dodawania zgłoszeń oraz adopcji.

Innym rozwiązaniem jest uruchomienie projektu w Visual Studio Code i wprowadzenie poniższych komend w terminalu:
npm install
npm run dev

W celu przetestowania funkcjonalności związanych z zalogowanymi użytkownikami poniżej zamieszczone zostaną przykładowe konta, na których można testować aplikację:

Konto osoby prywatnej:
login: user1234@wp.pl
hasło: 1234

Konto schroniska:
login: schronisko1@wp.pl
hasło: 1234