Dokumentacja projektu

Aplikacja internetowa dla gabinetu weterynaryjnego.

Autorzy: Michał Dybaś, Krystian Dybaś

Spis treści

[1. Informacje ogólne 3](#_Toc99443230)

[a. Wybrany sposób realizacji projektu 3](#_Toc99443231)

[b. Przeznaczenie projektu 4](#_Toc99443232)

[2. Schemat bazy danych – diagram ERD 5](#_Toc99443233)

[3. Struktura projektu 6](#_Toc99443234)

[4. Opis struktury projektu 7](#_Toc99443235)

[5. Wygląd aplikacji 9](#_Toc99443236)

# Informacje ogólne

## Wybrany sposób realizacji projektu

Projekt został zrealizowany w języku Java v.12 (Maven). Projekt został rozszerzony o biblioteki Spring Framework, SpringBoot, SpringBoot-Jpa, SpringBoot-Security, SpringBoot-Thymeleaf, Thymeleaf-Extras-Security, Lombok, SpringBoot-DevTools oraz sterownik dostępu do bazy danych Postgres i H2.

Strony internetowe zbudowane zostały przy użyciu hipertekstowego języka znaczników HTML wzbogaconego o szablony Thymeleaf, kaskadowych arkuszy stylów CSS, biblioteki Bootstrap (biblioteka CSS) i skryptów JavaScript.

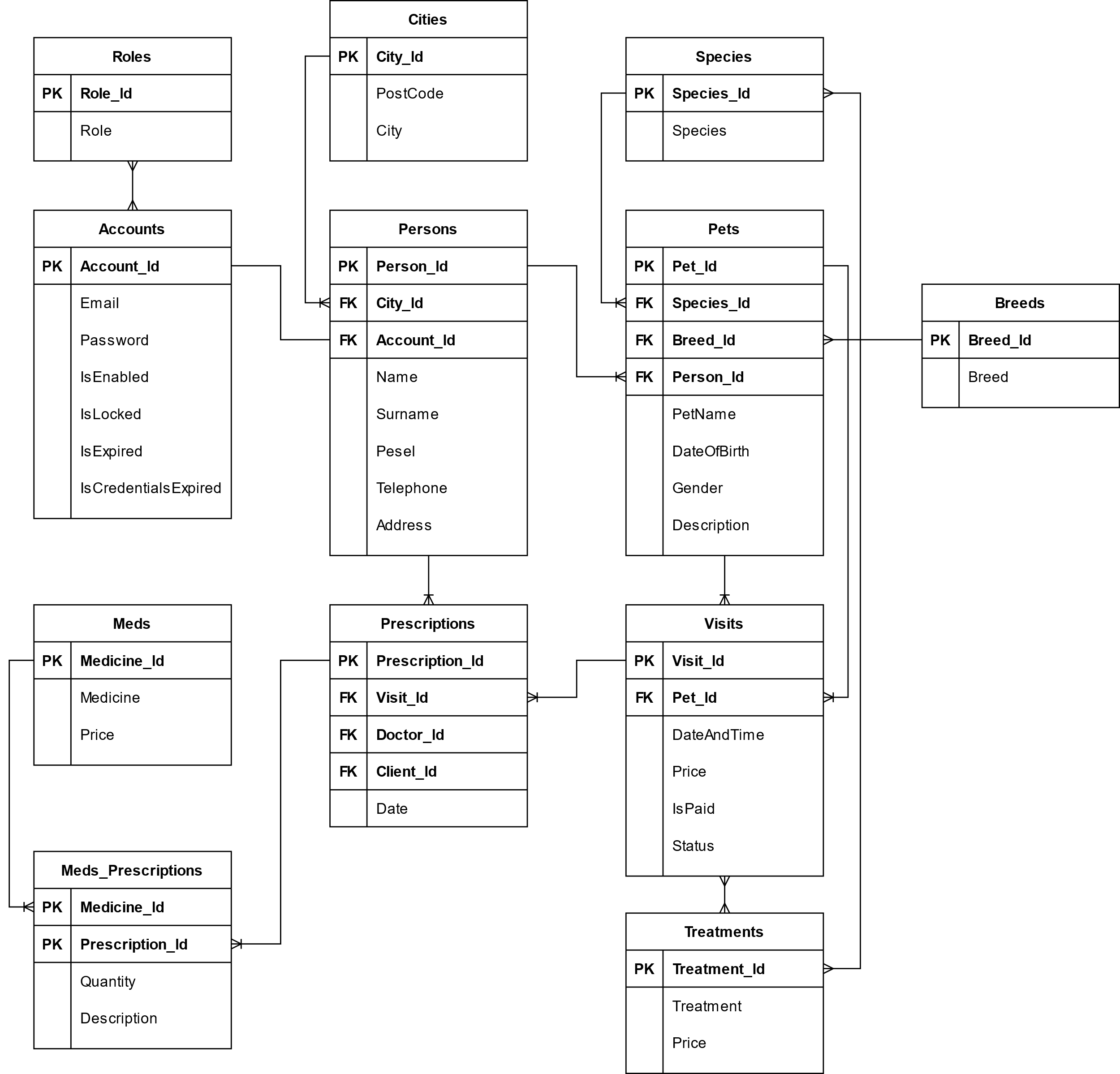
Projekt wykorzystuje wzorzec architektoniczny MVC (Model-View-Controller), który rozdziela projekt na 3 główne moduły:

* **Model** – jest pewną reprezentacją problemu bądź logiki aplikacji. Zmiana danych przez użytkownika odbywa się za pomocą kontrolera, ten modyfikuje model, model odświeża widok, a informacja z widoku dociera do użytkownika.
* **Widok** – opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika. Może składać się z podwidoków (fragmentów) odpowiedzialnych za mniejsze części interfejsu.
* **Kontroler** – przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków.

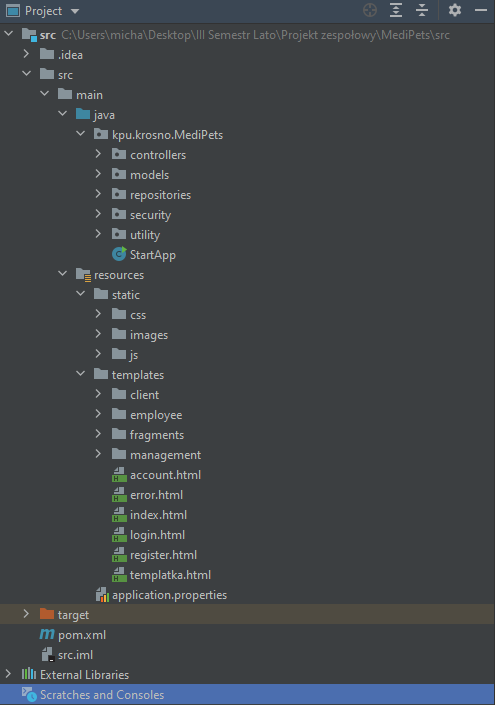
## Przeznaczenie projektu

Przeznaczeniem projektu jest wspomaganie pracy gabinetu weterynaryjnego, poprzez przechowywanie informacji na temat gabinetu i wszystkich klientów oraz udostępnienie interfejsu po stronie pracownika i klienta.

# Schemat bazy danych – diagram ERD



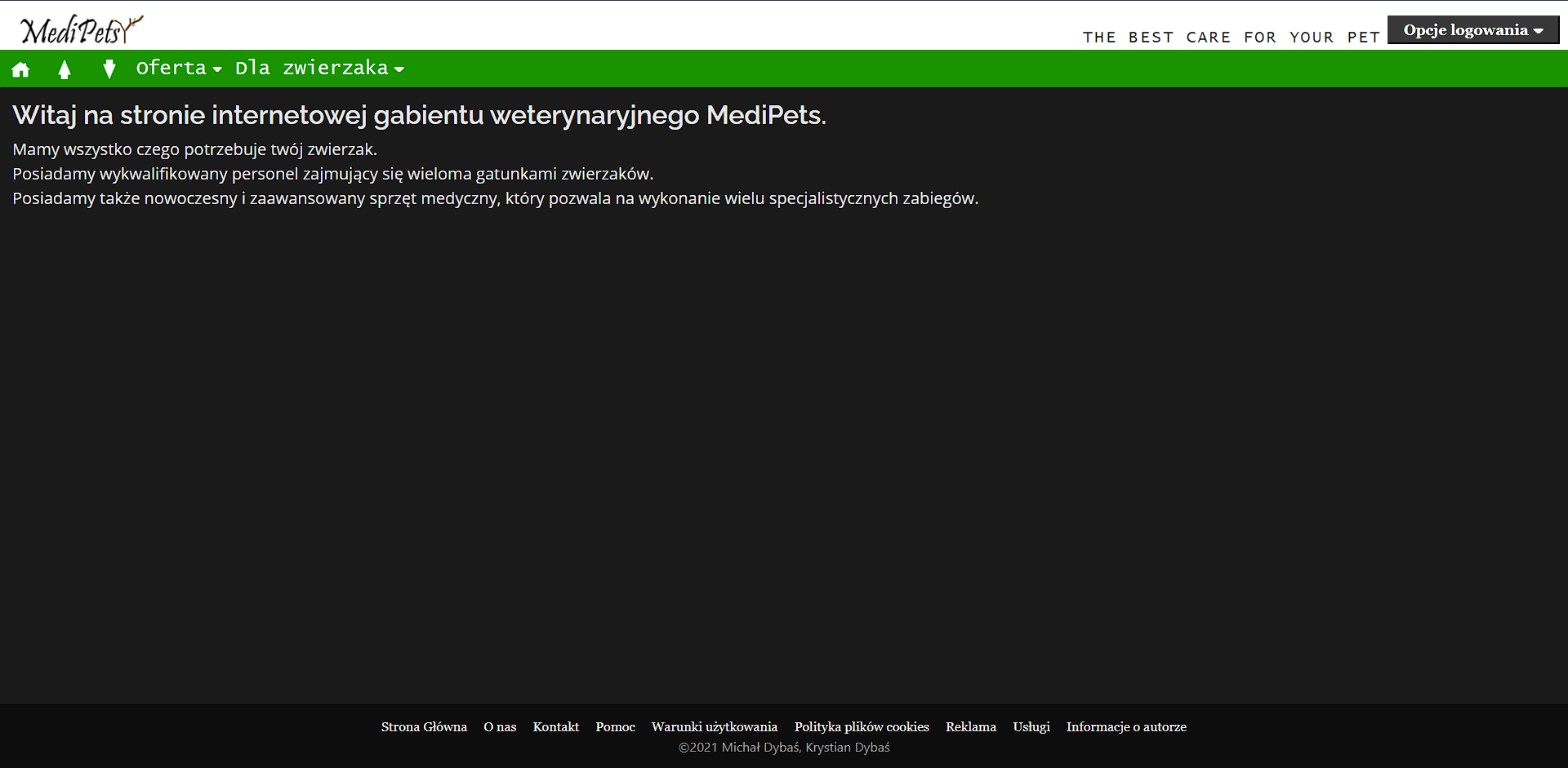
# Struktura projektu



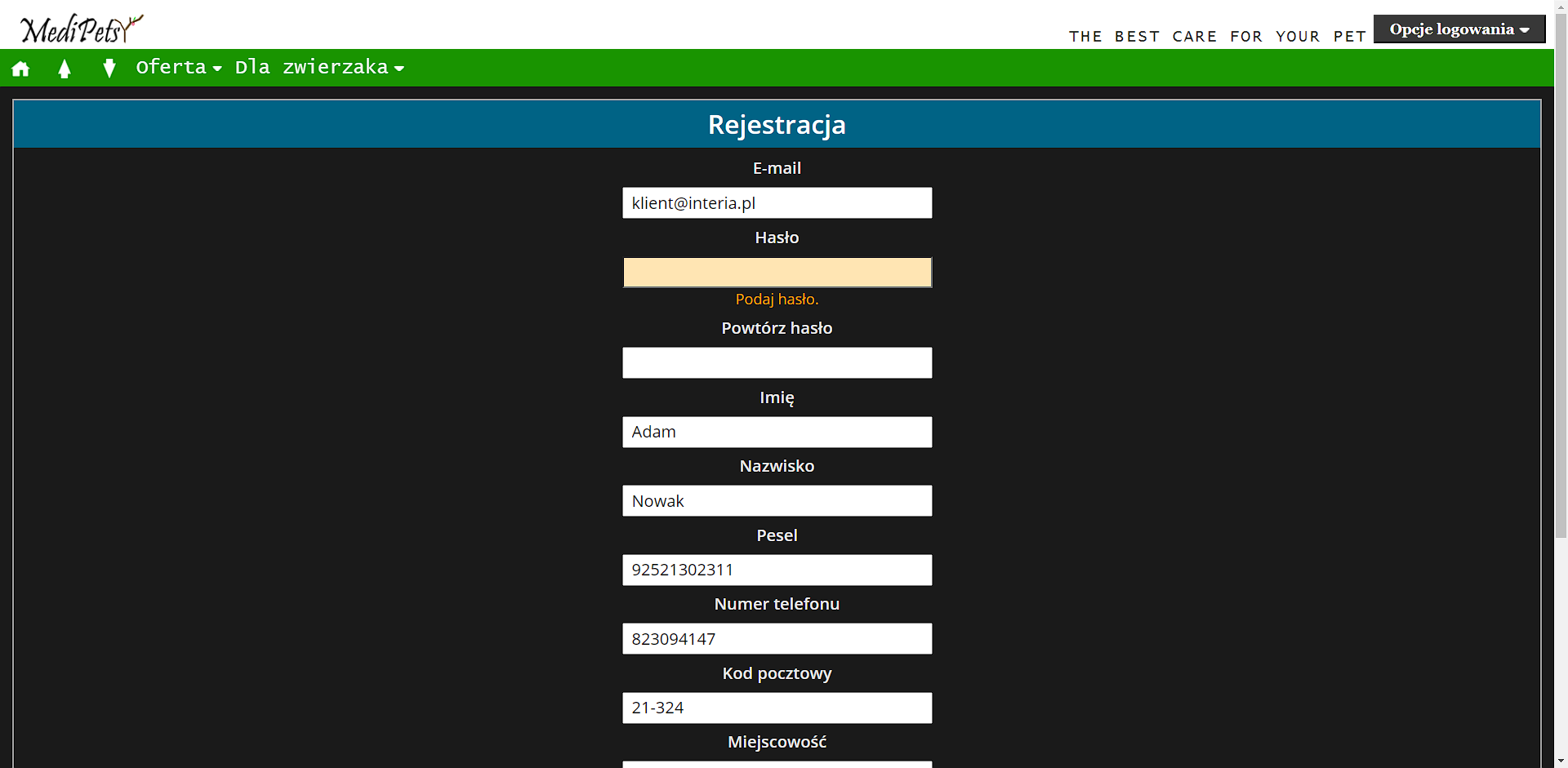
# Opis struktury projektu

* Folder Java zawiera główny pakiet aplikacji kpu.krosno.MediPets, który składa się na:
  + Klasę *StartApp.java* zawierającą metodę startową naszej aplikacji,
  + Pakiet *controllers –* zawierający klasy kontrolerów zajmujące się obsługą żądań http/https,
  + Pakiet *models* – zawierający klasy modeli reprezentujących i definiujących tabele w bazie danych i ich relacje. Tabele w bazie danych i ich relacje za pomocą mechanizmu ORM są tworzone i aktualizowane na podstawie modeli,
  + Pakiet *repositories* – zawierający klasy repozytoriów, które udostępniają interfejsy do wykonywania zapytań do bazy danych, dzięki wykorzystaniu mechanizmu ORM,
  + Pakiet *security* – zawierający klasy bezpieczeństwa umożliwiające weryfikację i autoryzację użytkowników. Klasy zawierają te między innymi definicję sesji użytkownika, definicję adresów url dostępnych dla danej roli użytkownika itd,
  + Pakiet *utility* – zawierający klasy obiektów wykorzystywanych do pobierania i wstawiania danych do formularzy oraz klasę CustomValidator udostępniającą metody do walidacji pól formularzy.
* Folder resources zawiera statyczne i dynamiczne elementy witryny takie jak strony html, arkusze stylów itd.:
  + Folder *static/css* - zawiera arkusze stylów css wykorzystywane w aplikacji,
  + Folder *static/images* - zawiera grafiki wykorzystywane w aplikacji,
  + Folder *static/js* - zawiera skrypty JavaScript wykorzystywane w aplikacji,
  + Folder *templates* – zawiera główne strony html witryny takie jak logowanie rejestracja i strona główna, a także pod foldery zawierające strony html dla danych ról,
  + Folder *templates/fragments* – zawiera fragmenty stron html menu, stopkę, oraz konfigurację zawierającą sekcję head,
  + Foldery *templates/clients, templates/employee, templates/managment* – zawierają strony html kolejno dla ról klienta, pracownika i administratora,
  + Plik application.properties – zawiera konfigurację aplikacji.

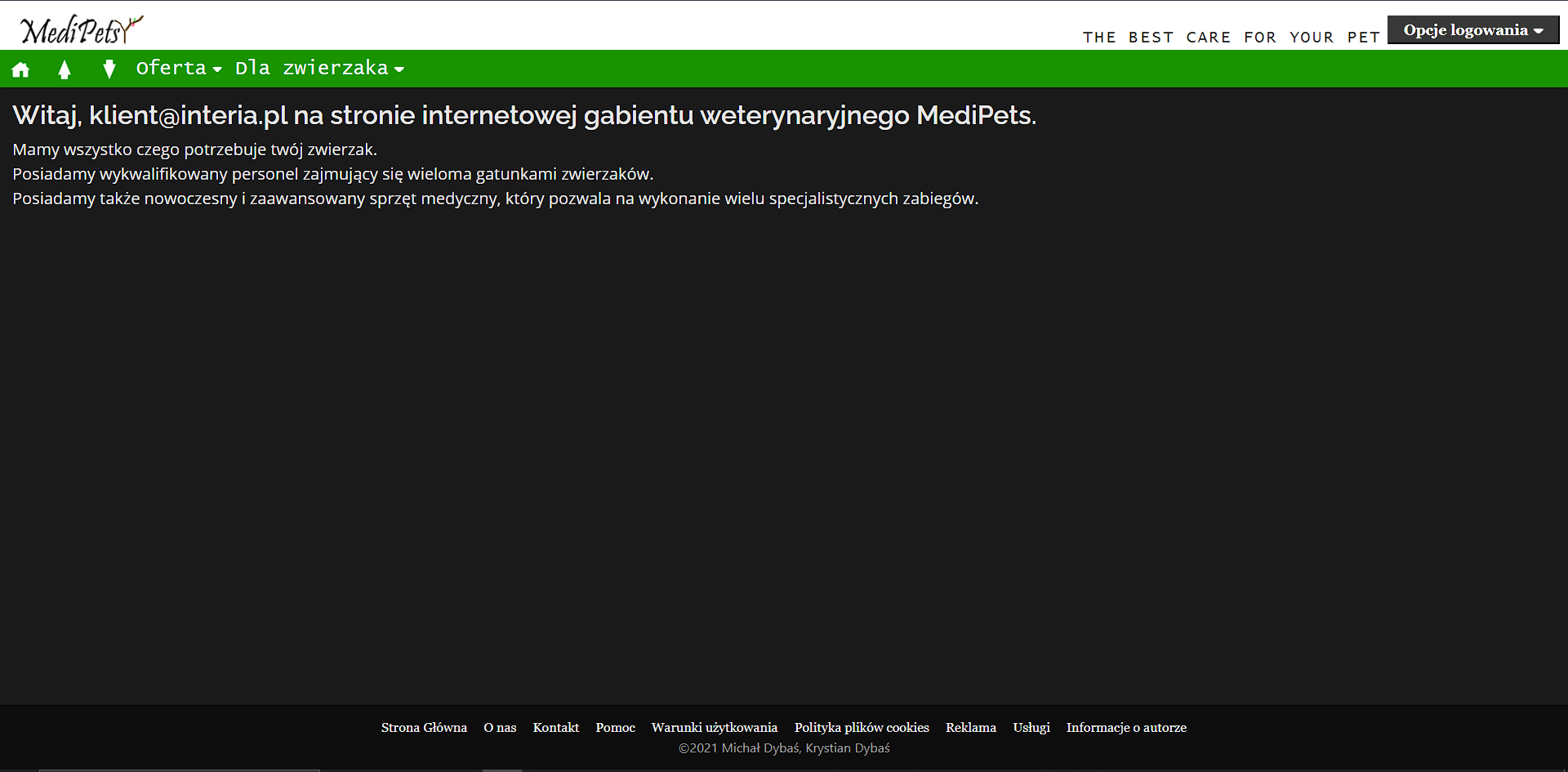
# Wygląd aplikacji



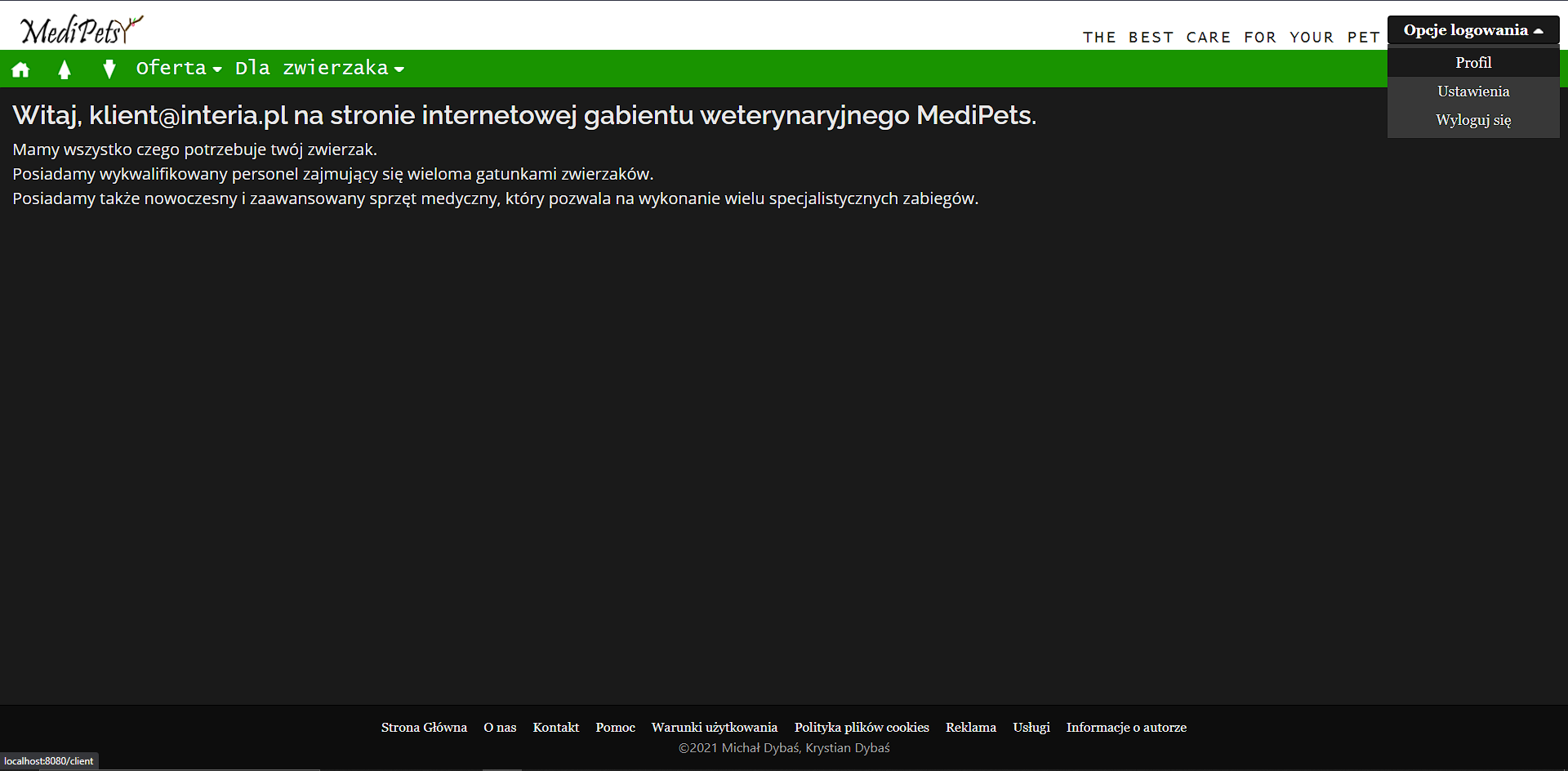
Rysunek Strona główna



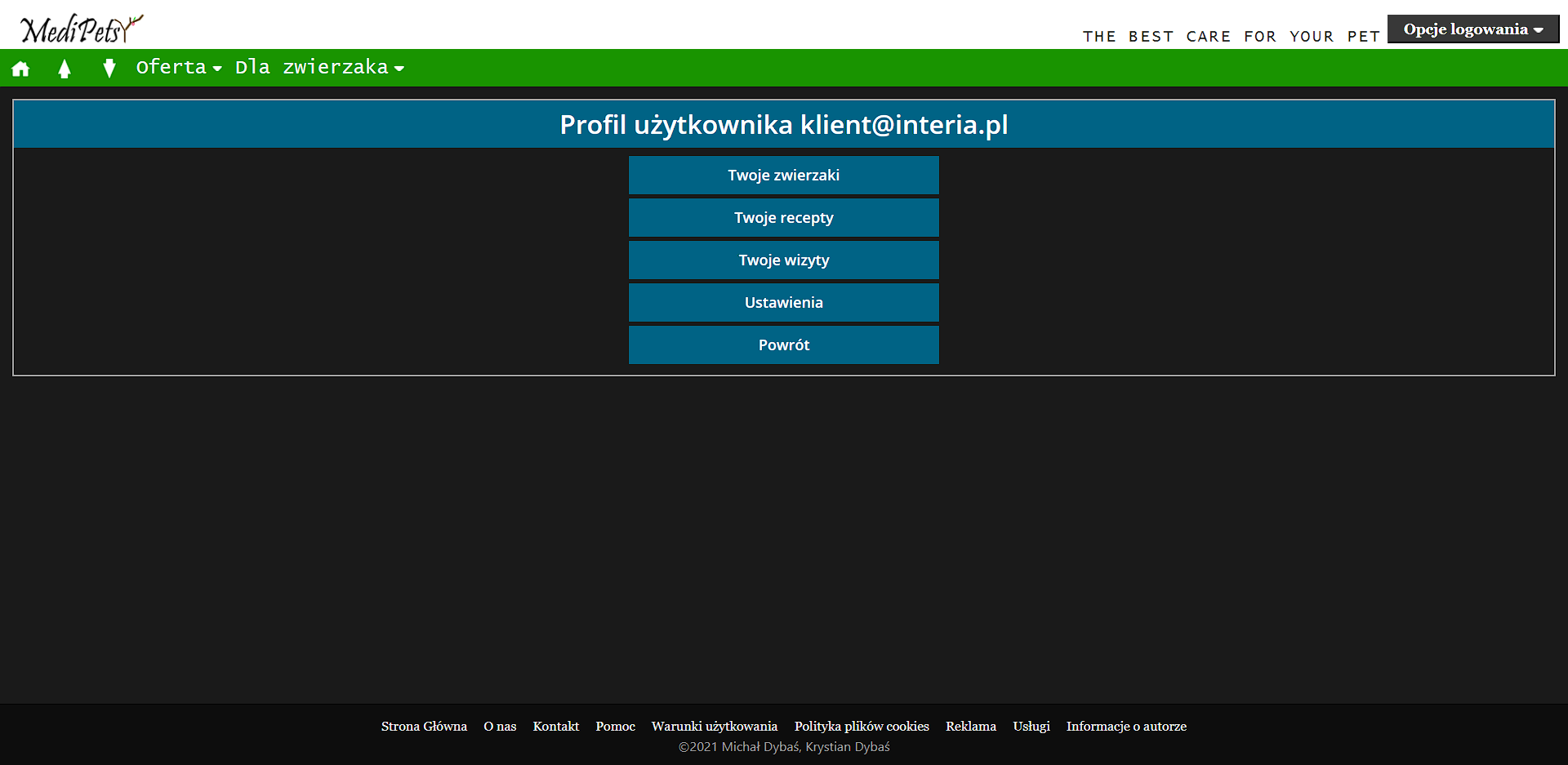
Rysunek Formularz rejestracyjny - walidacja formularzy



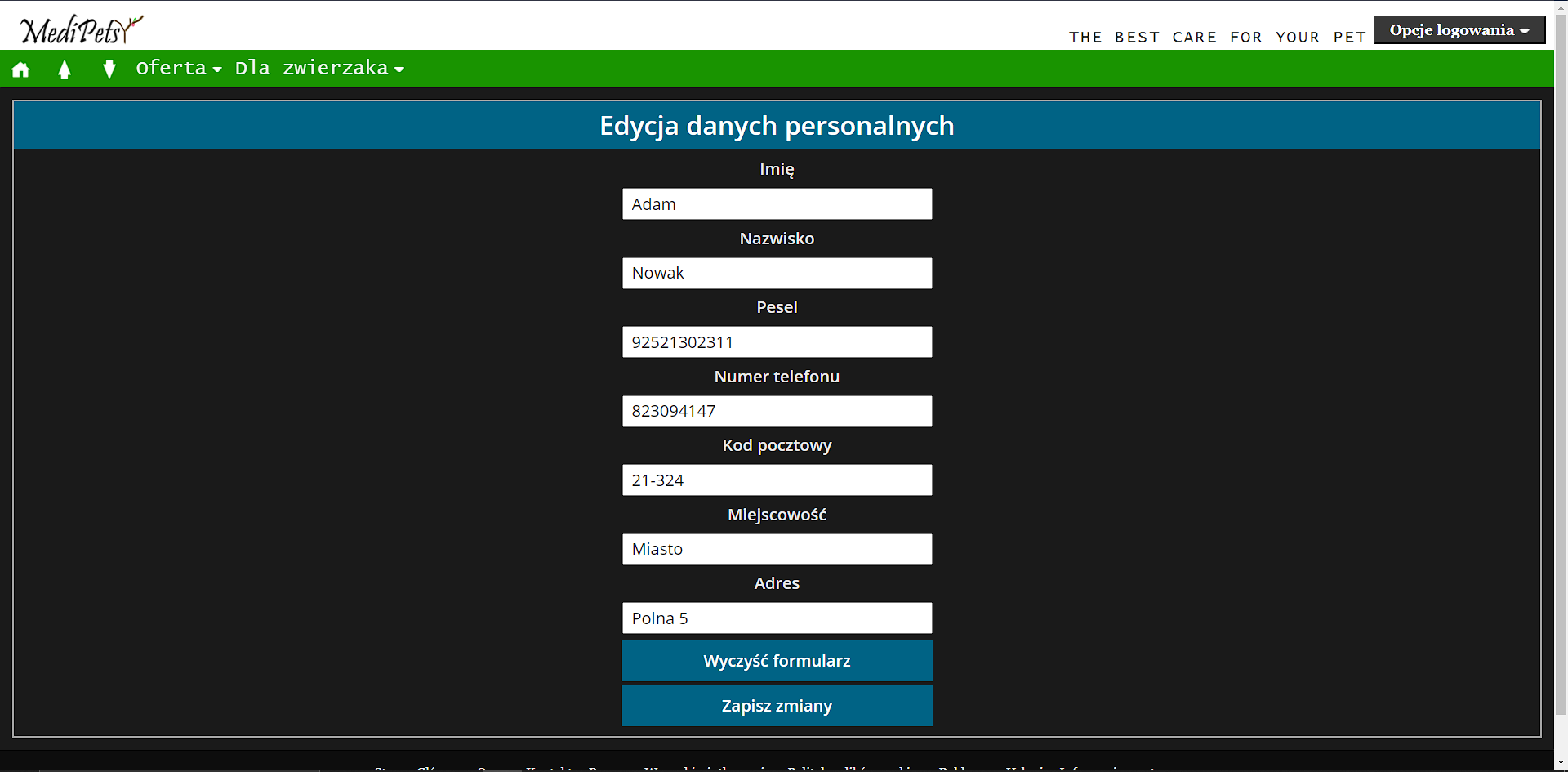
Rysunek Okno po udanej rejestracji - logowanie na nowe konto następuje automatycznie



Rysunek Zalogowany użytkownik ma dostęp do swojego profilu, ustawień konta (zmiana danych personalnych) oraz do opcji wylogowania się.



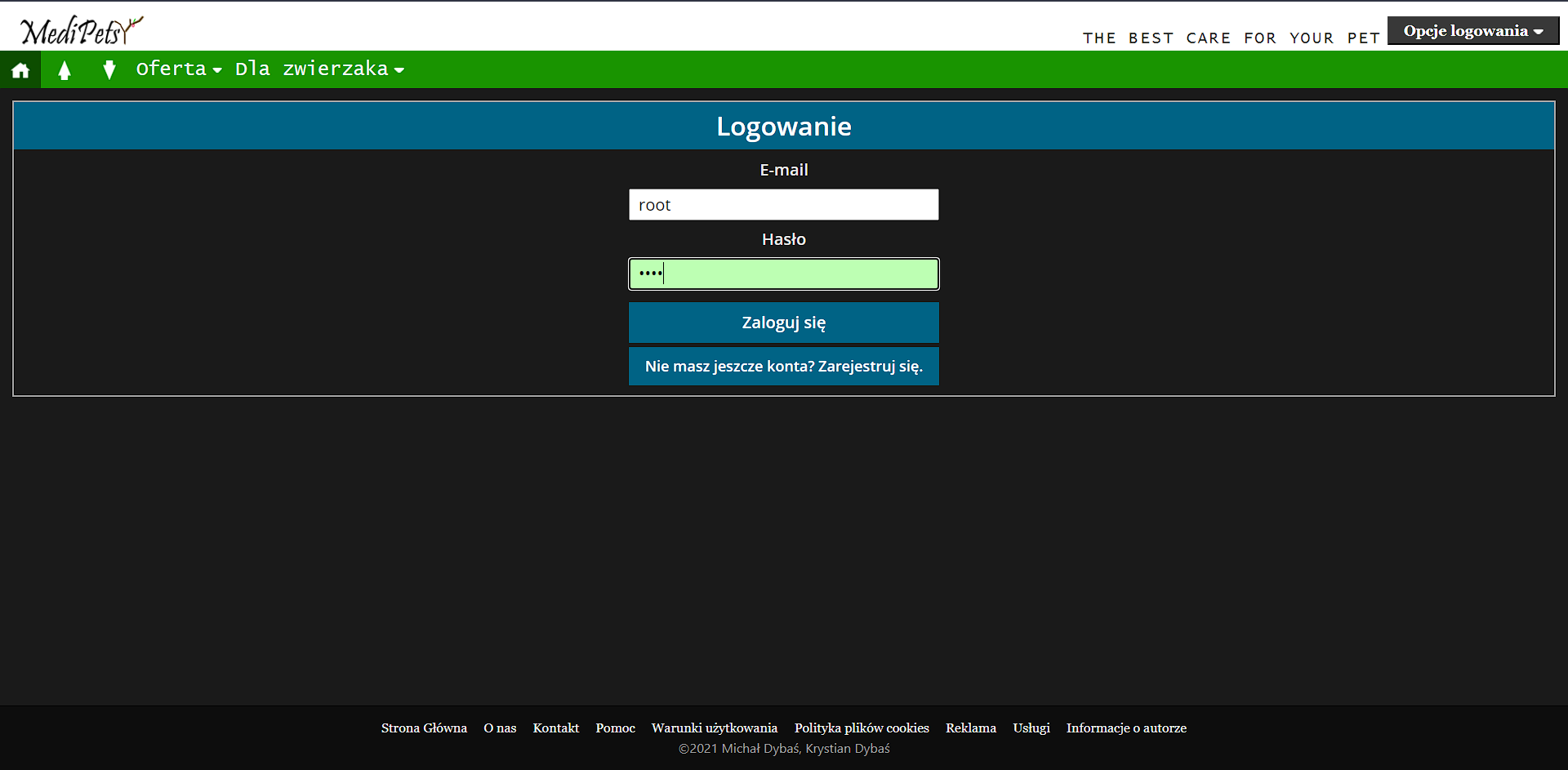
Rysunek Profil klienta



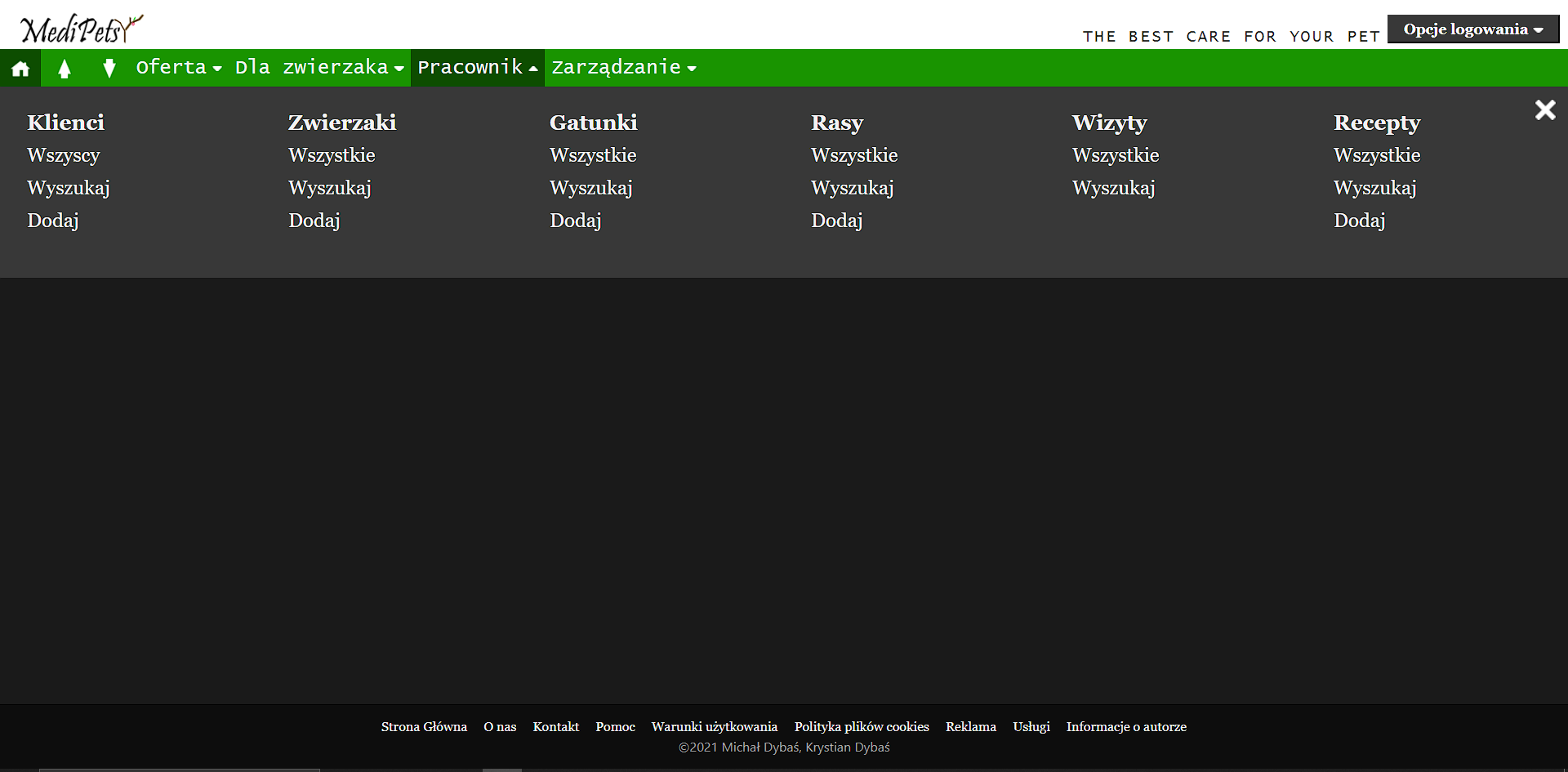
Rysunek Ustawienia konta



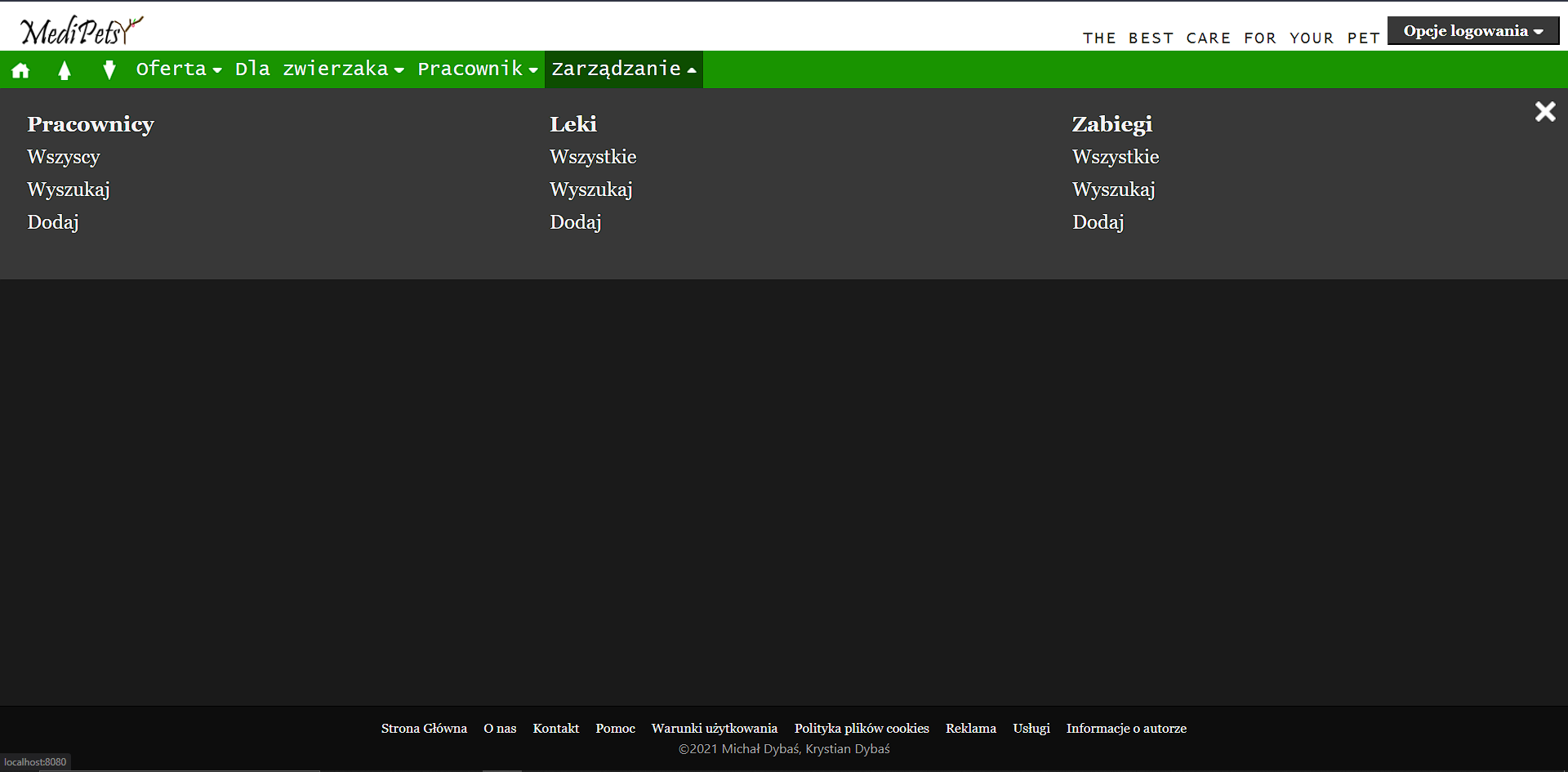
Rysunek Jak widać nowo utworzony klient nie posiada dodanych żadnych dodatkowych danych



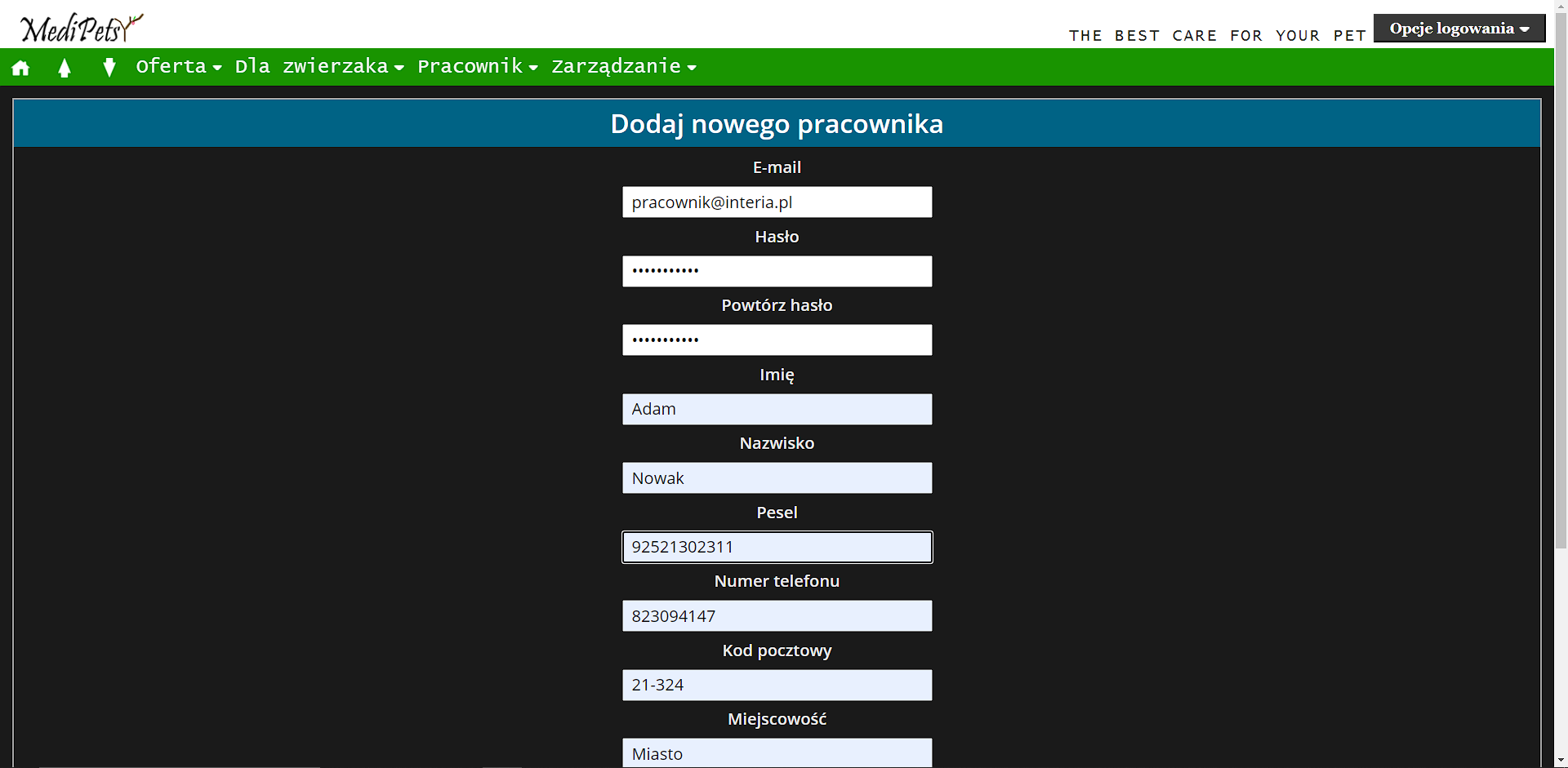
Rysunek Okno logowania się na konto super użytkownika, po uprzednim wylogowaniu się z konta klienta



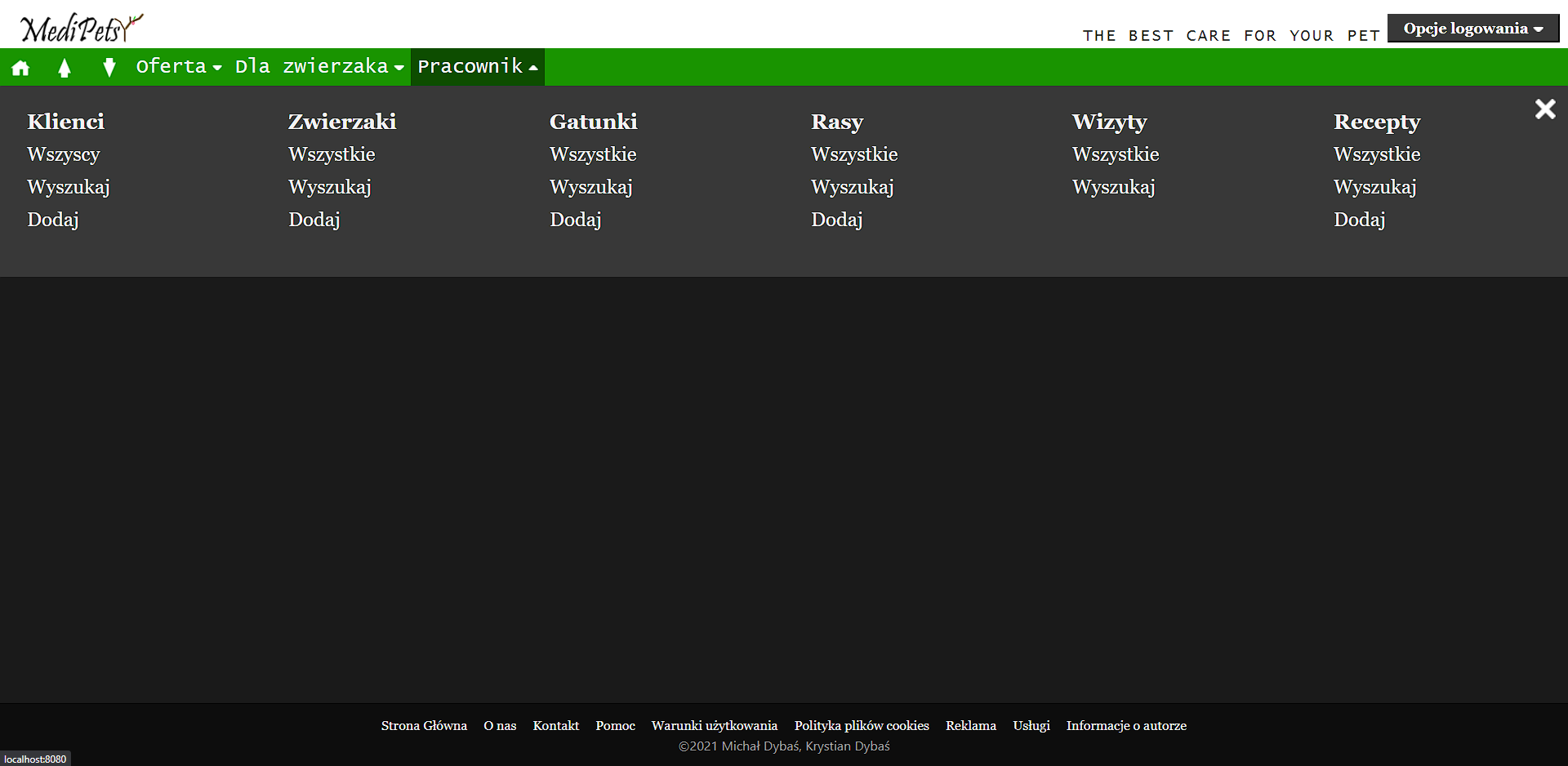
Rysunek Jak widać dla konta superużytkownika dostępne są wszystkie opcje modyfikacji i dodawania danych



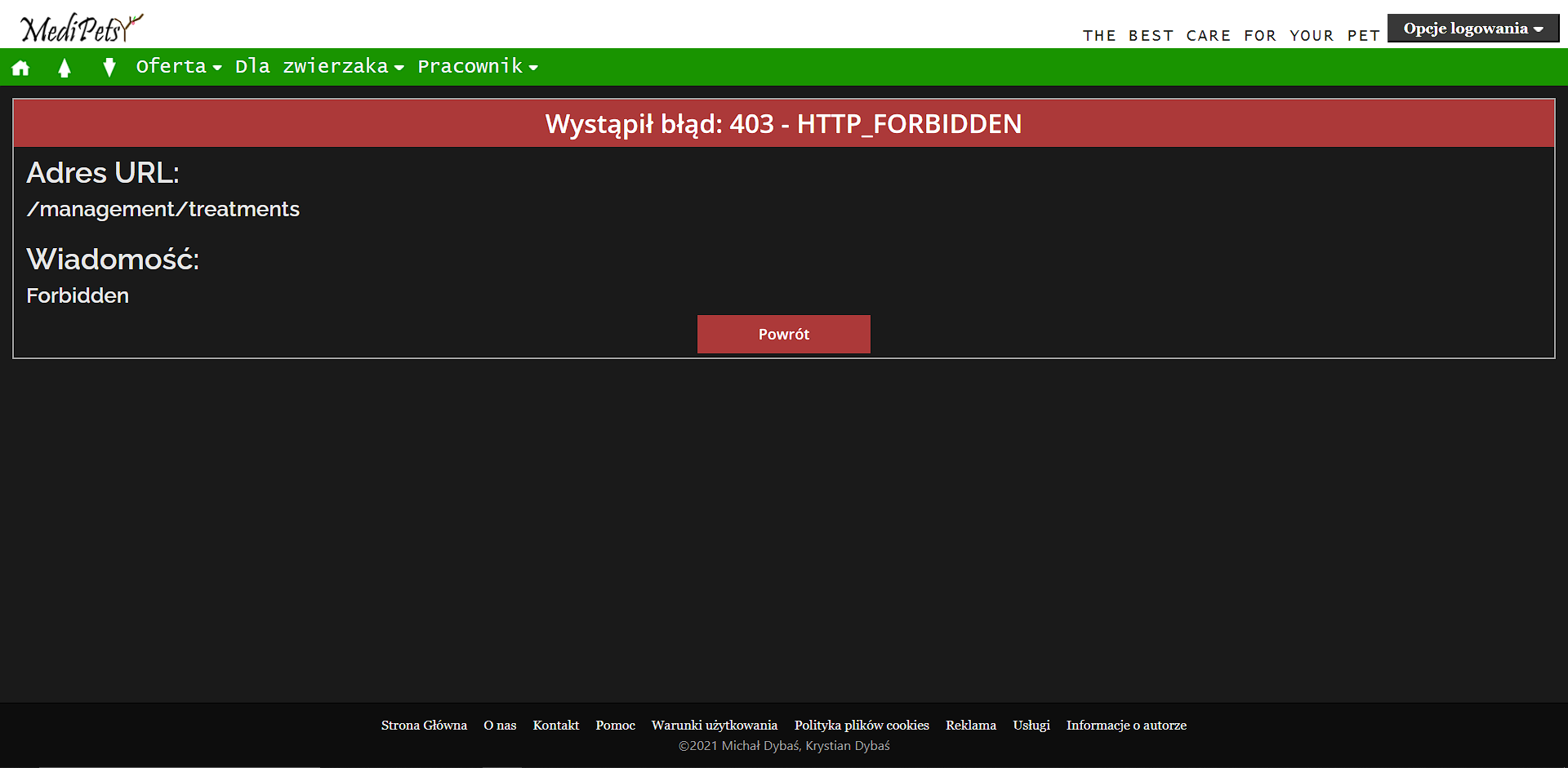
Rysunek Jak widać dla konta superużytkownika dostępne są wszystkie opcje modyfikacji i dodawania danych



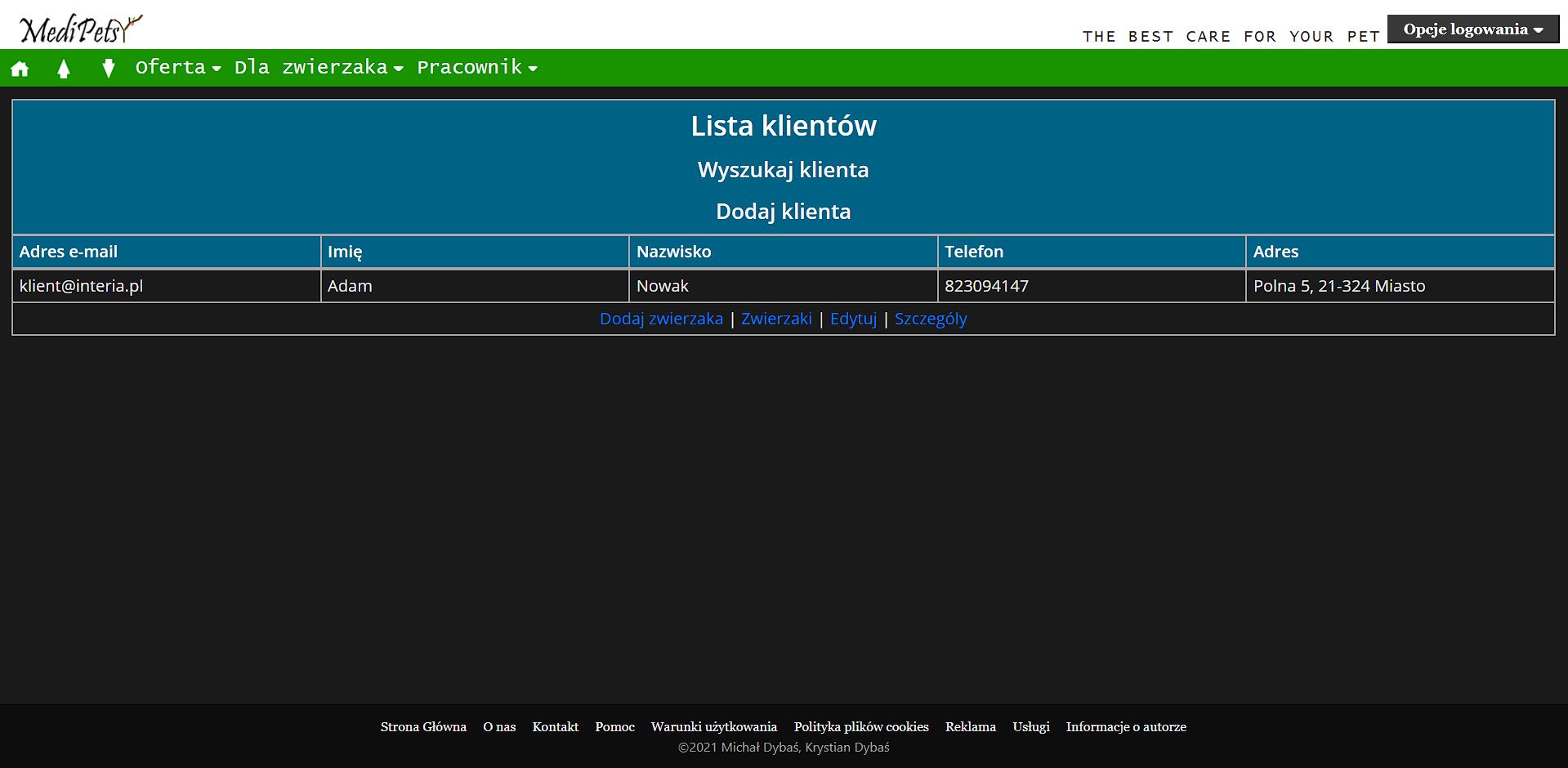
Rysunek Dodajmy nowego pracownika z konta superużytkownika



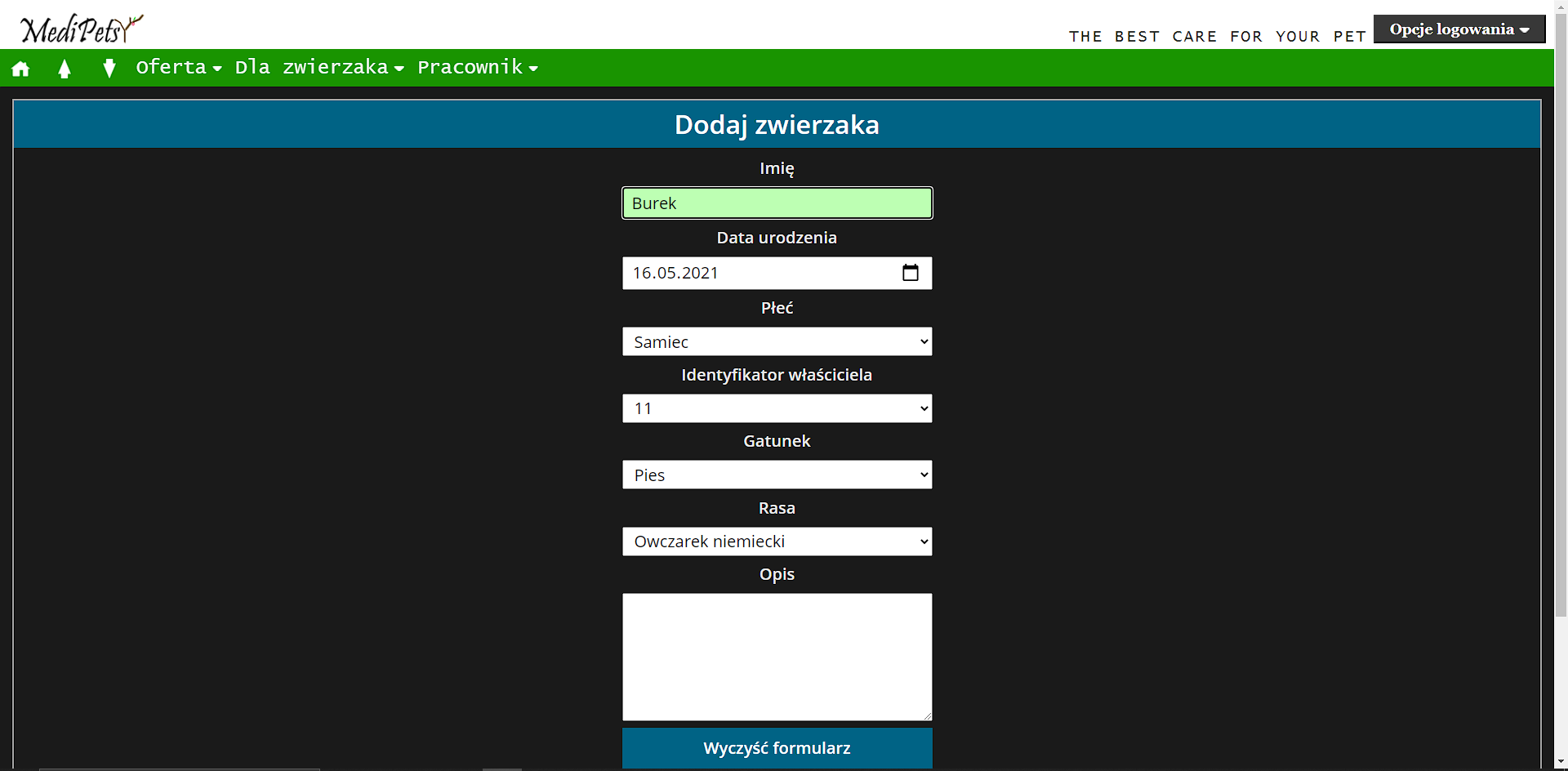
Rysunek Po zalogowaniu się na konto administratora jak widać opcje zarządzania są niedozwolone



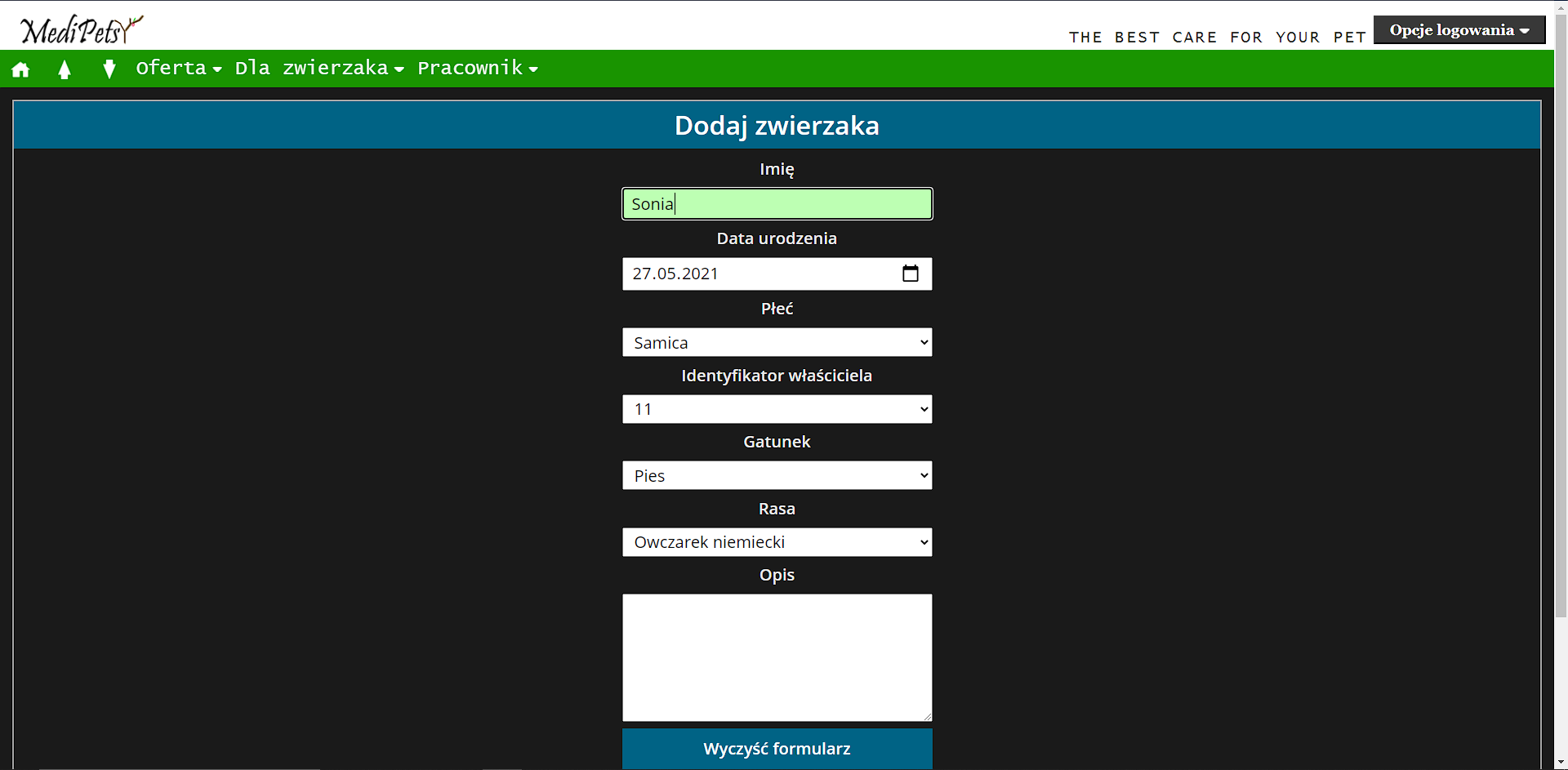
Rysunek Próba dostania się na stronę zarezerwowaną dla superużytkownika z konta pracownika spotka się z odmową ze strony aplikacji



Rysunek Teraz po przejściu na listę klientów dostępno z poziomu menu Pracownik, możemy dodać do klienta zwierzaka



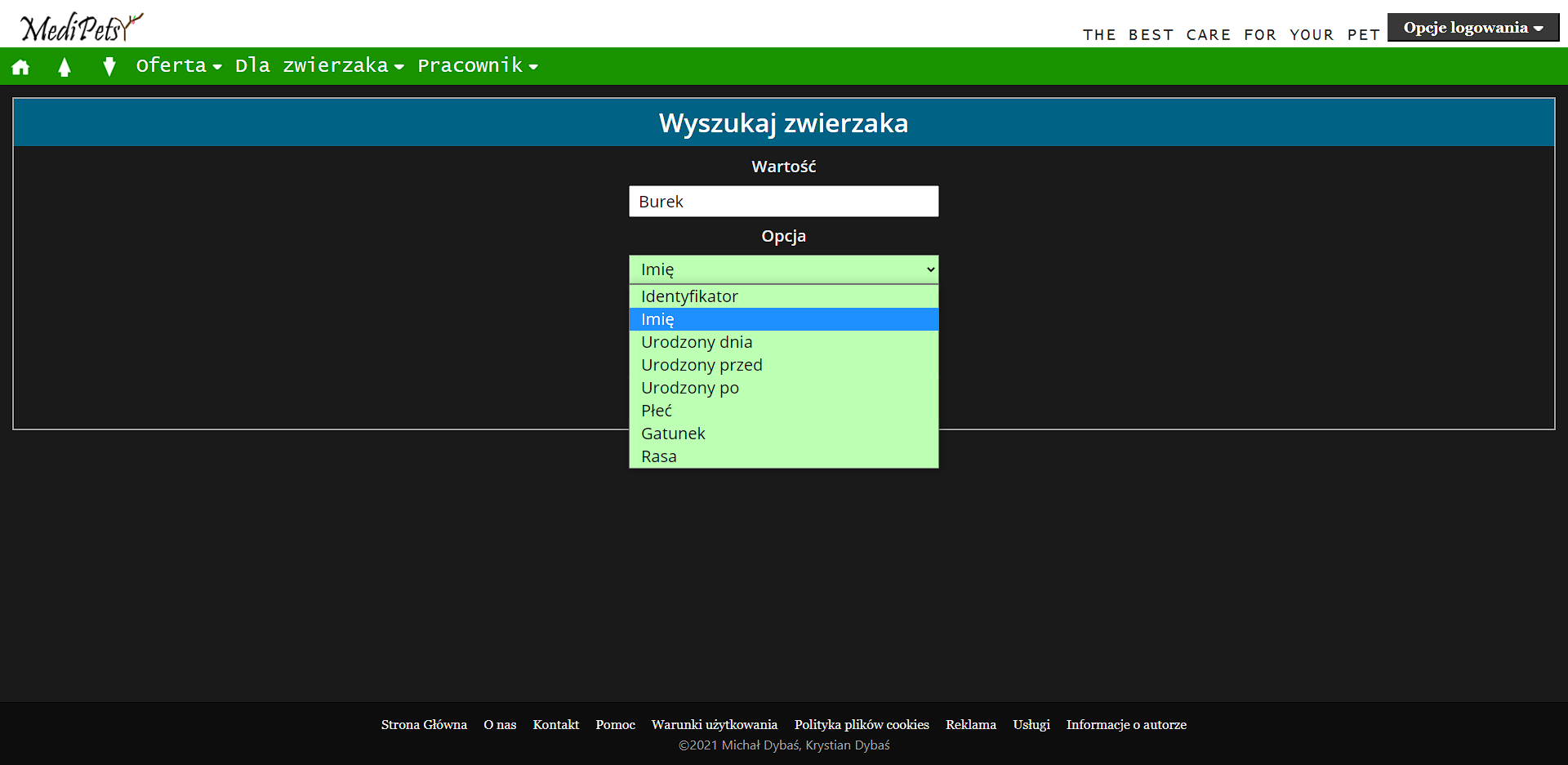
Rysunek Okno dodania zwierzaka



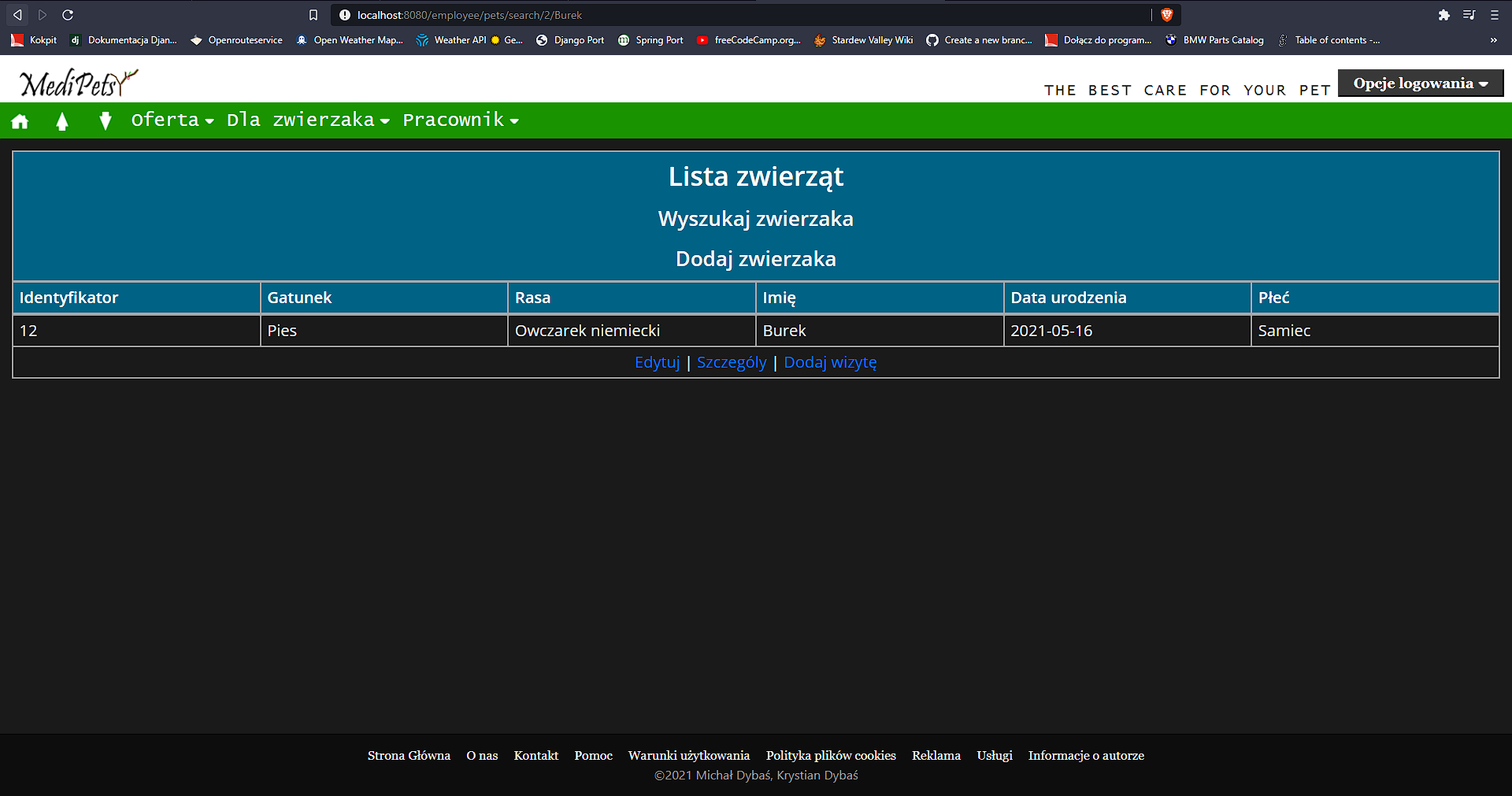
Rysunek Do naszego klienta dodamy sobie jeszcze drugiego zwierzaka w celu pokazania działania wyszukiwarki



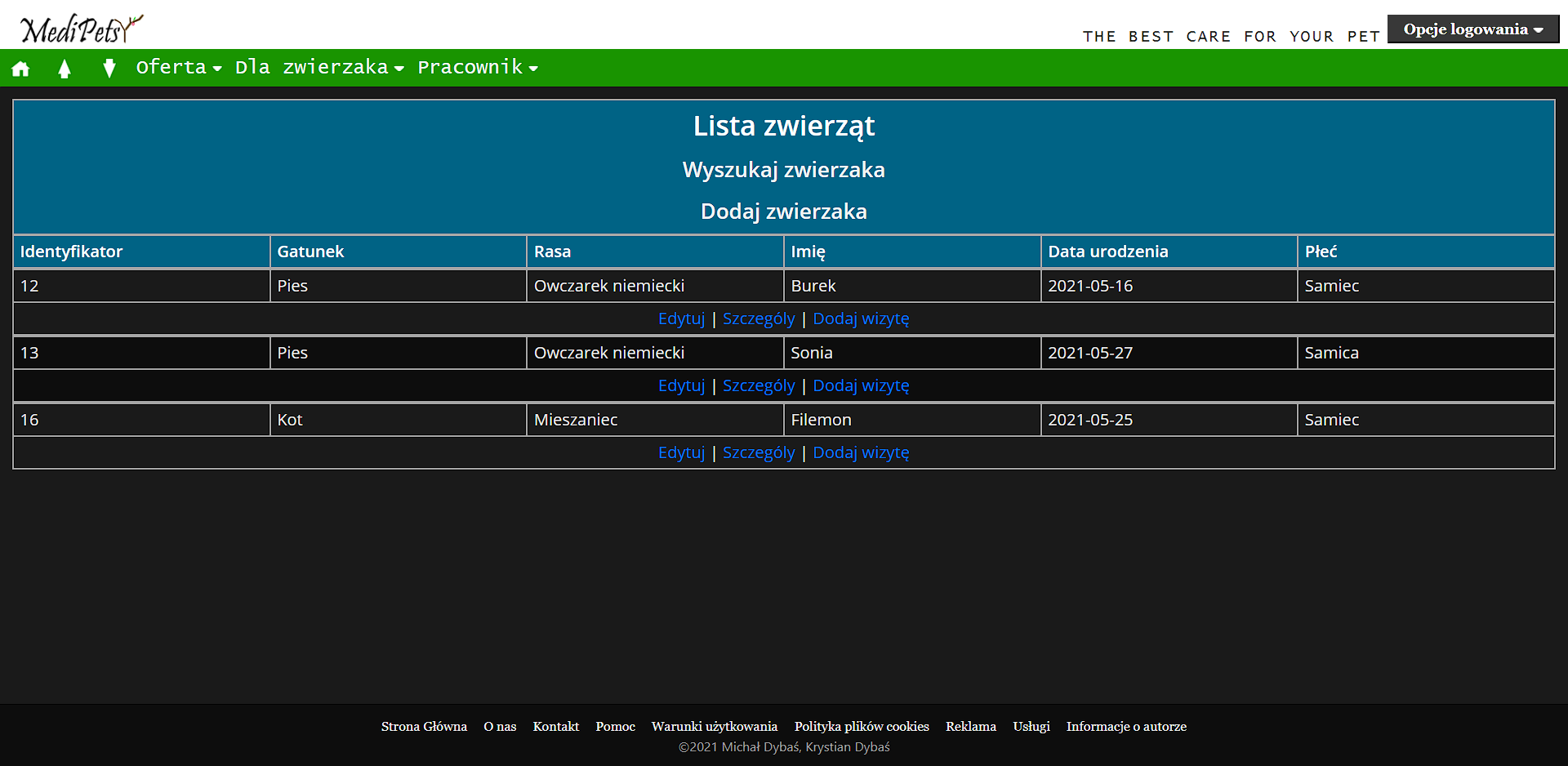
Rysunek Lista zwierzaków prezentuje się następująca



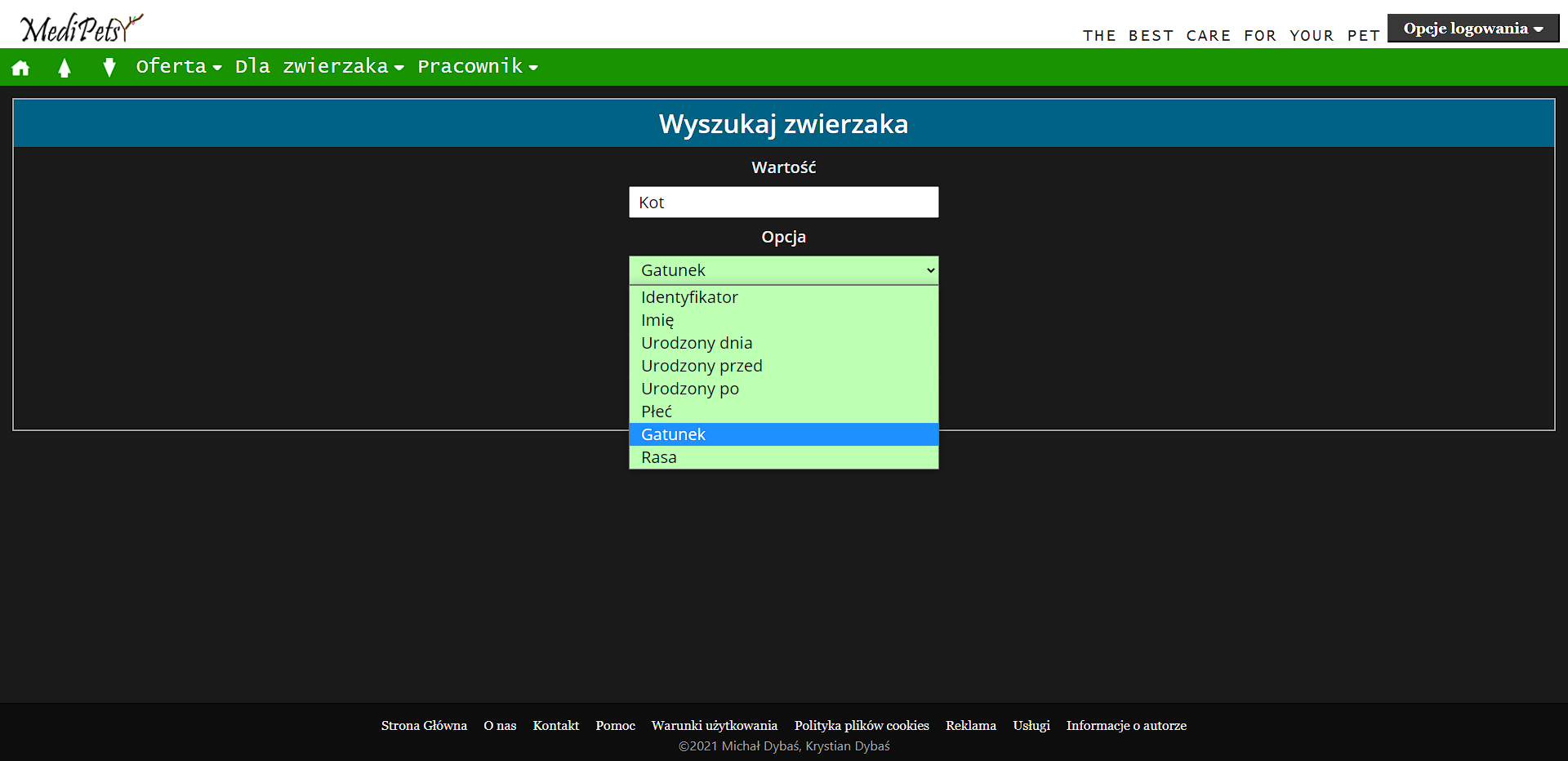
Rysunek Spróbujemy wyszukać sobie teraz psa o imieniu Burek



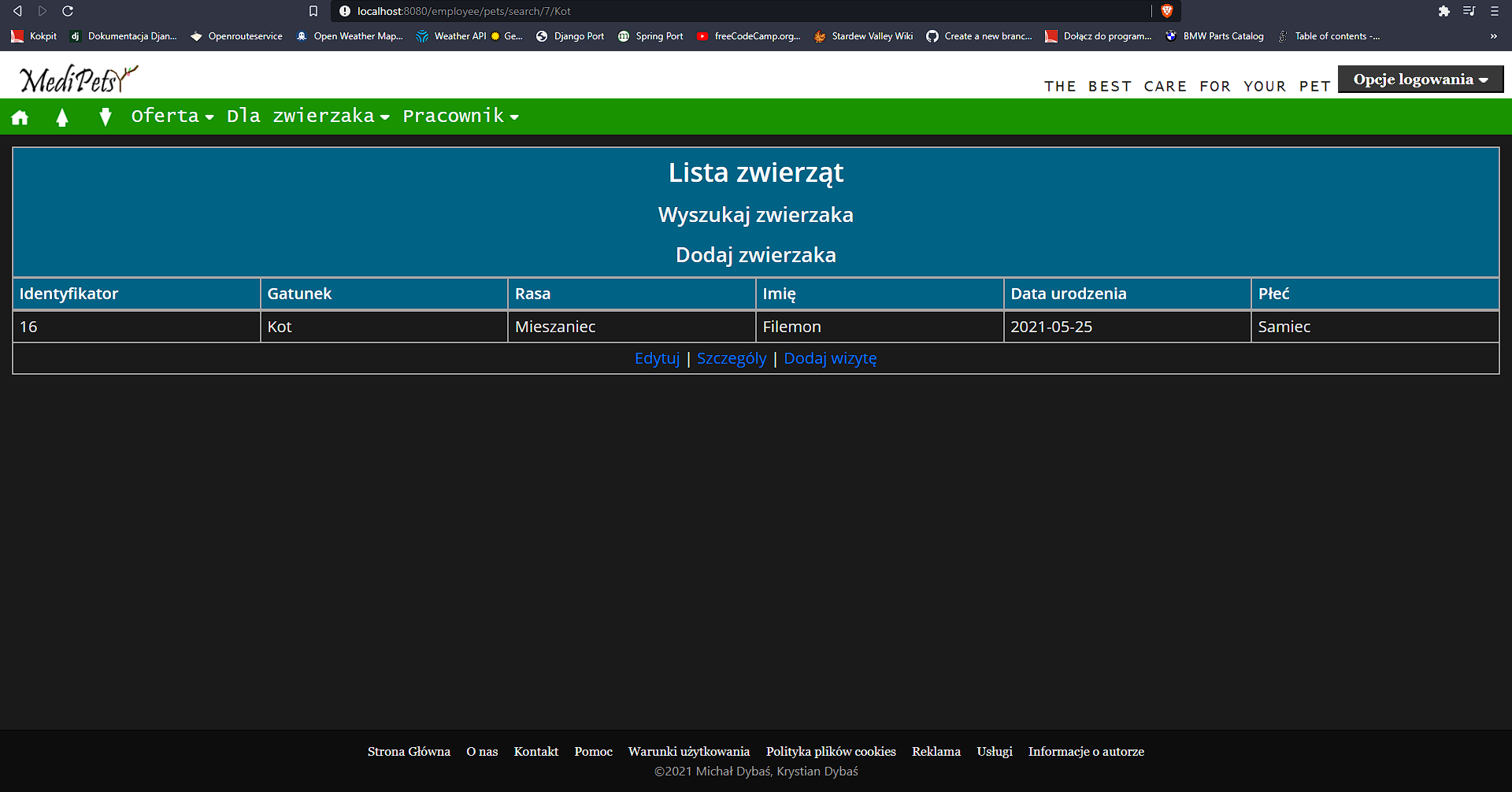
Rysunek Jak widać operacja wyszukiwania zwróciła nam prawidłowy wynik



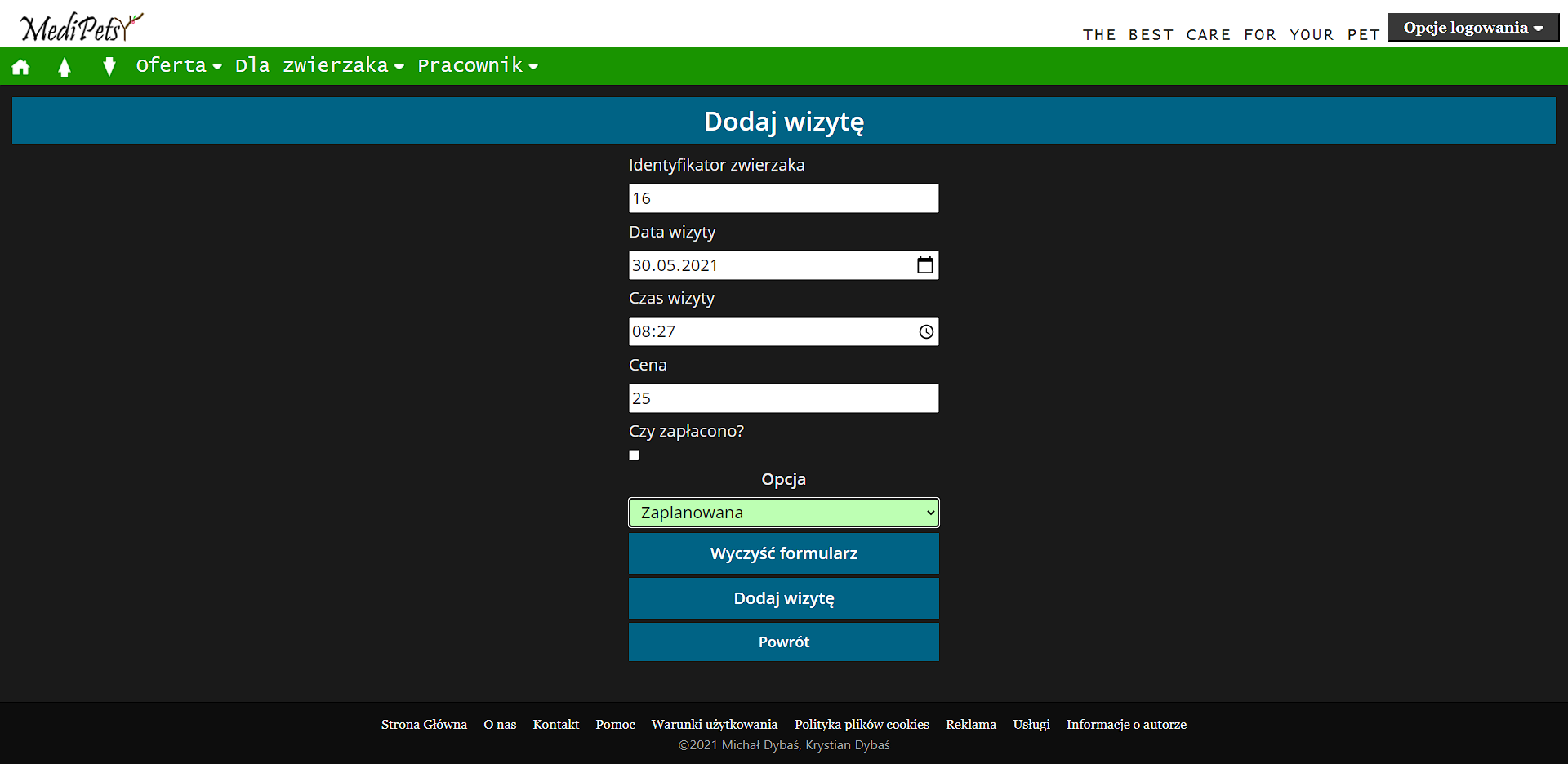
Rysunek Dodałem jeszcze zwierzaka, jakiego gatunkiem jest kot



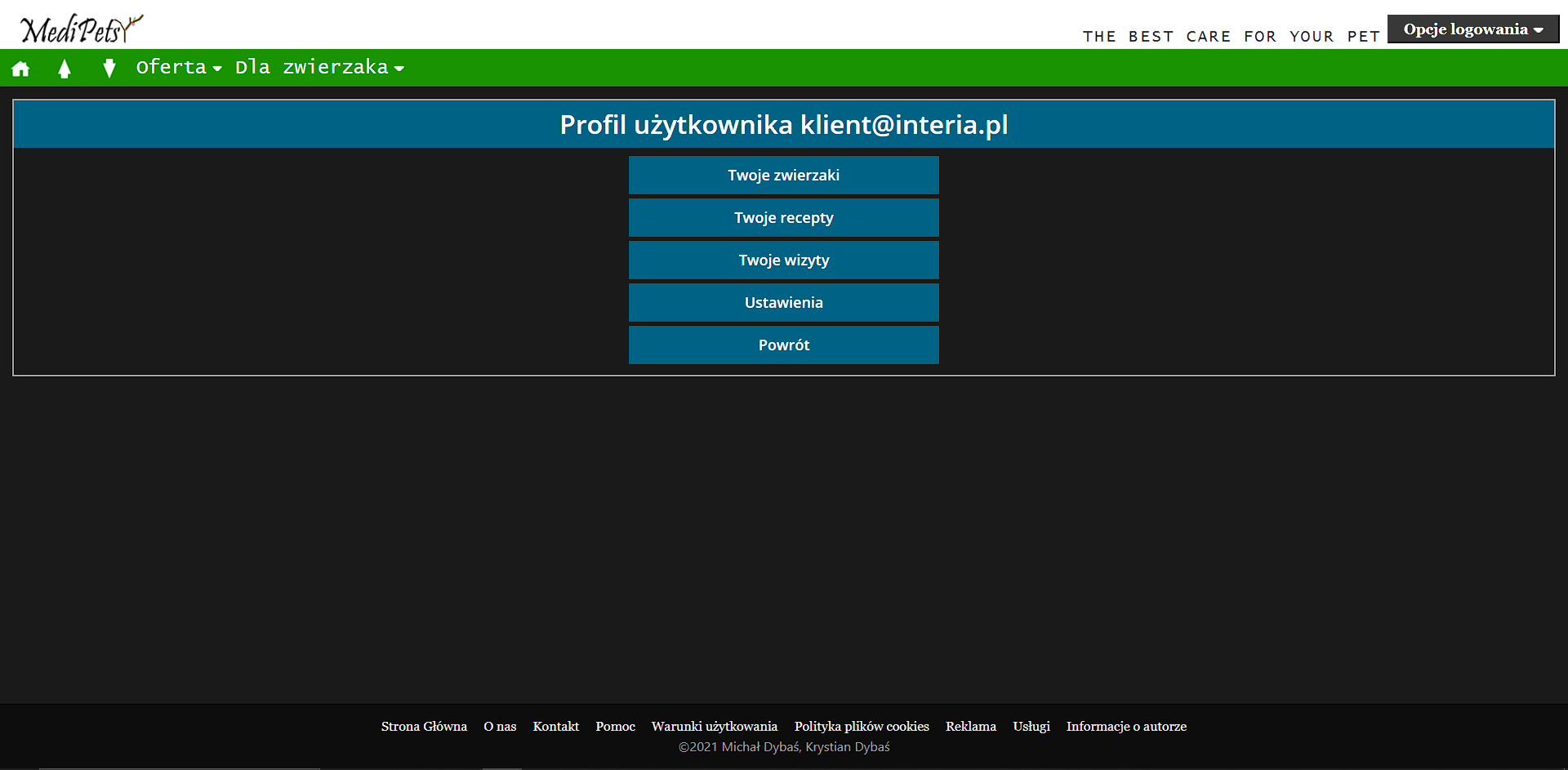
Rysunek Wyszukajmy teraz wszystkie zwierzaki rasy kot



Rysunek Wynik został zwrócony prawidłowo



Rysunek Klikając opcje dodaj wizyte przeszliśmy do okna dodania wizyta dla Filemona



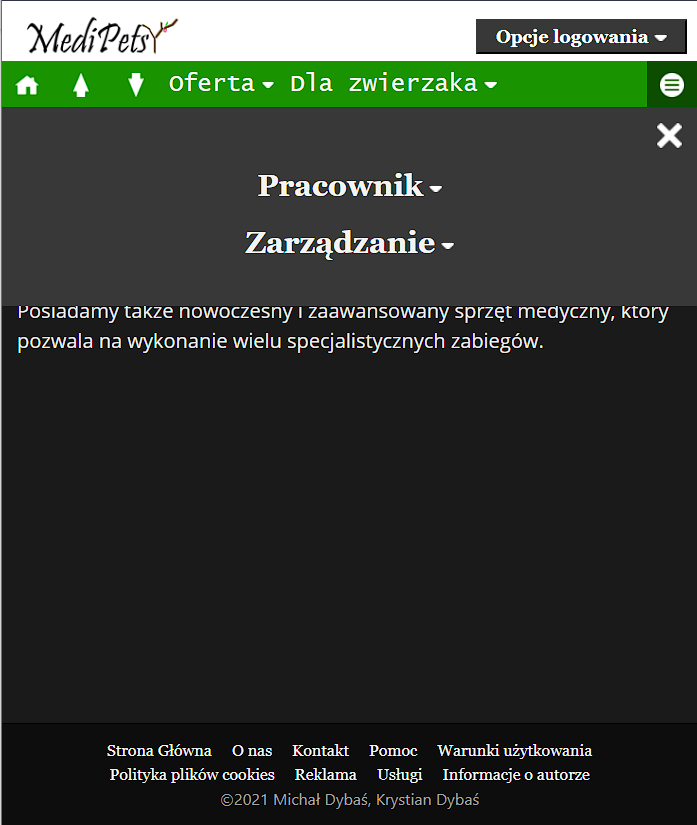
Rysunek Po zwierzaków do klienta zobaczymy teraz na jego profil



Rysunek Lista zwierzaków danego klienta, Filemon nie został pokazany, ponieważ przypisaliśmy mu innego właściciela jakim jest pracownik



Rysunek Strona jest w pełni responsywna, po zalogowaniu się np. na konto root i zmniejszeniu szerokości ekranu, pojawia nam się rozwijane menu zawierające wszystkie dostępne kategorie



Rysunek Jeśli ekran pomieści niektóre opcje, to do menu rozwijanego trafiają tylko te które nie zmieszczą się na głównym pasku