

**Aplikacja “NeighBro”**

**Projekt kompetencyjny**

**Grzegorz Książczyk**

**Mateusz Pabich**

Spis treści

[Założenia projektu: 3](#_Toc2089253453)

[Stack technologiczny: 3](#_Toc1984196091)

[Baza danych: 3](#_Toc1355962869)

[Backend: 3](#_Toc1433360497)

[Frontend: 3](#_Toc1991245819)

[Opis funkcjonalności: 3](#_Toc388907487)

[Strona logowania 4](#_Toc904235468)

[Panel administratora: 5](#_Toc1542884432)

[Strona startowa 5](#_Toc976123670)

[Menu 6](#_Toc794100524)

[Kreator kont 6](#_Toc1767645316)

[Kreator osiedla 9](#_Toc1959165218)

[Dane użytkownika 16](#_Toc537993226)

[Zarządzanie opłatami 17](#_Toc120392900)

[Ogłoszenia 19](#_Toc1460639115)

[Głosowania 20](#_Toc1262018859)

[Obsługa usterek 22](#_Toc1292730418)

[Raporty 24](#_Toc1630858504)

[Dokumentacja API 25](#_Toc2095440492)

[Panel właściciela 26](#_Toc2126479388)

[Menu 27](#_Toc584945888)

[Ogłoszenia 28](#_Toc683133048)

[Rachunki 29](#_Toc1889799690)

[Głosowania 30](#_Toc1191368992)

[Zgłoś usterkę 31](#_Toc925343987)

[Baza danych 33](#_Toc521640174)

[Opis 34](#_Toc1920322063)

[Schemat bazy danych: 34](#_Toc737848032)

[Opis tabel w bazie danych 35](#_Toc110179656)

[Dane użytkowników: 35](#_Toc1513426574)

[Dane o głosowaniach: 35](#_Toc1688247315)

[Dane o budynkach: 36](#_Toc473789161)

[Dane o przestrzeniach: 36](#_Toc1735467620)

[Dane o umowach najmu: 37](#_Toc2089997333)

[Dane o mieszkańcach: 37](#_Toc490066411)

[Dane o incydentach: 37](#_Toc1954030349)

[Dane o serwisantach: 38](#_Toc1297047278)

[Dane o fakturach i opłatach: 38](#_Toc1670787594)

[Backend 38](#_Toc491427768)

[Opis 39](#_Toc1323117406)

[Obsługa bazy danych 39](#_Toc1970072657)

[Wykorzystanie programowania defensywnego 40](#_Toc1795375551)

# Założenia projektu:

Aplikacja webowa do zarządzania wspólnotą mieszkaniową.

# Stack technologiczny:

## Baza danych:

* MySQL
* Docker

## Backend:

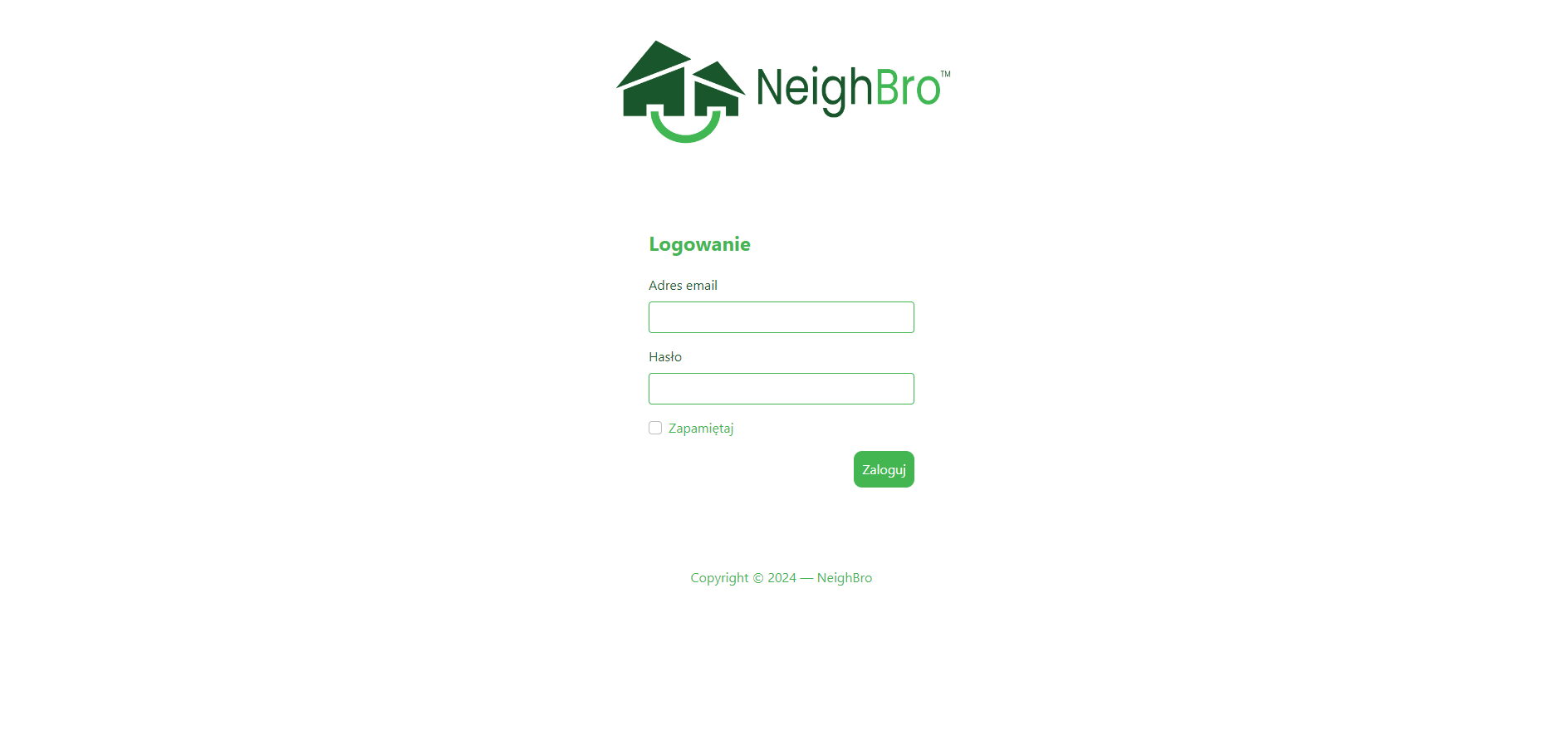
* FastAPI
* Python3 z następującymi pakietami:
  + httpx
  + uvicorn
  + fastapi
  + sqlalchemy
  + pymysql
  + cryptography
  + Iso4217

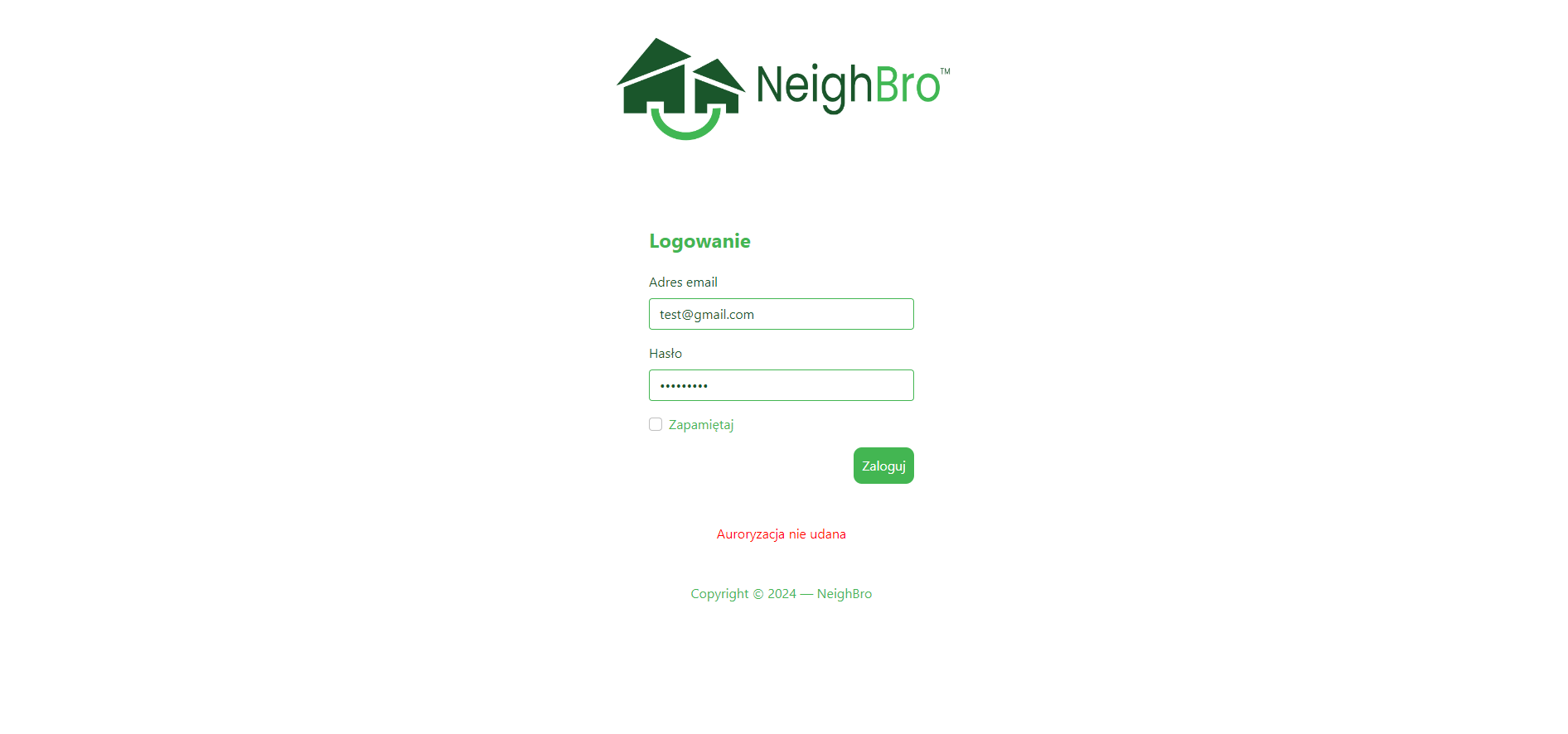
## Frontend:

* Bootstrap
* Javascript

# Opis funkcjonalności:

## Strona logowania

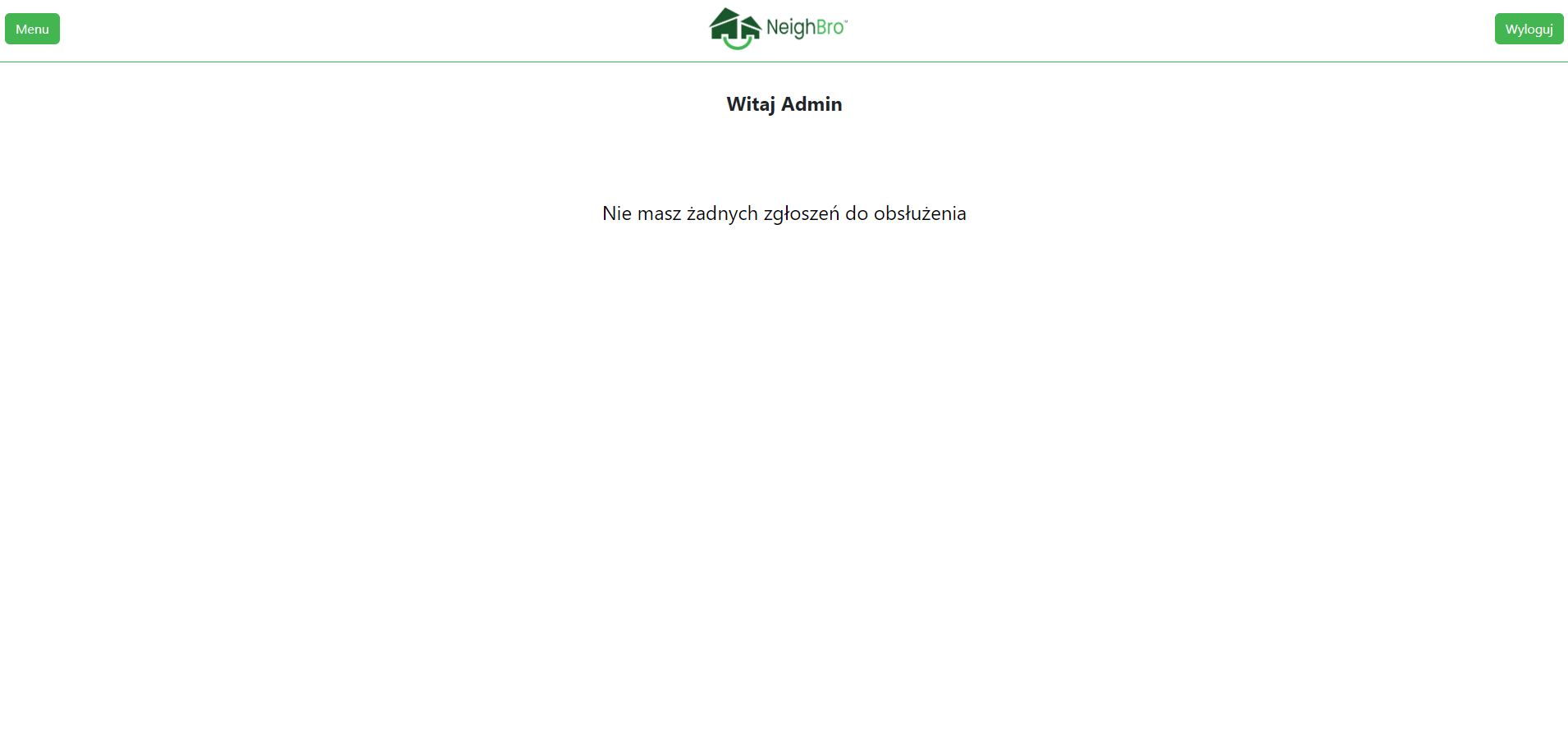
Pozwala na zalogowanie się do serwisu poprzez podanie nazwy użytkownika i hasła.

W przypadku nie przejścia autoryzacji zostanie wyświetlony komunikat  


Po zalogowaniu zostaniemy przeniesieni na panel administratora lub panel właściciela w zależności od typu naszego konta.

## Panel administratora:

### Strona startowa



Po zalogowaniu się na konto administratora zostanie nam wyświetlony powyższy widok zawierający:

-Przycisk menu

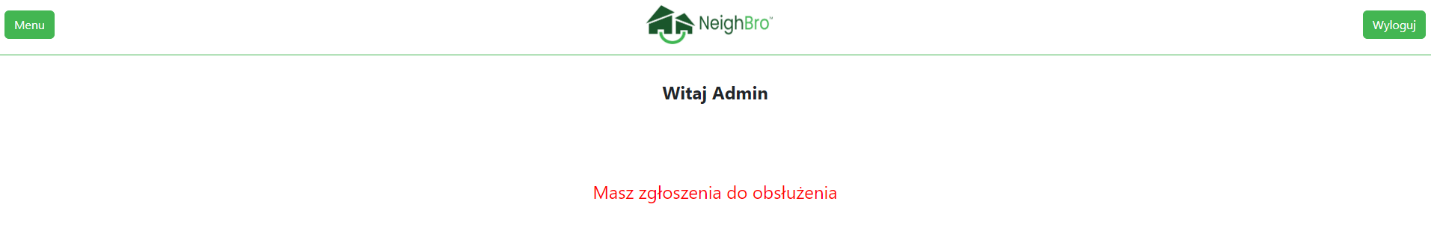
-Logo które jest jednocześnie odnośnikiem do strony startowej

-Przycisk wylogowania który wyloguje użytkownika i zmienia widok na stronę startowa

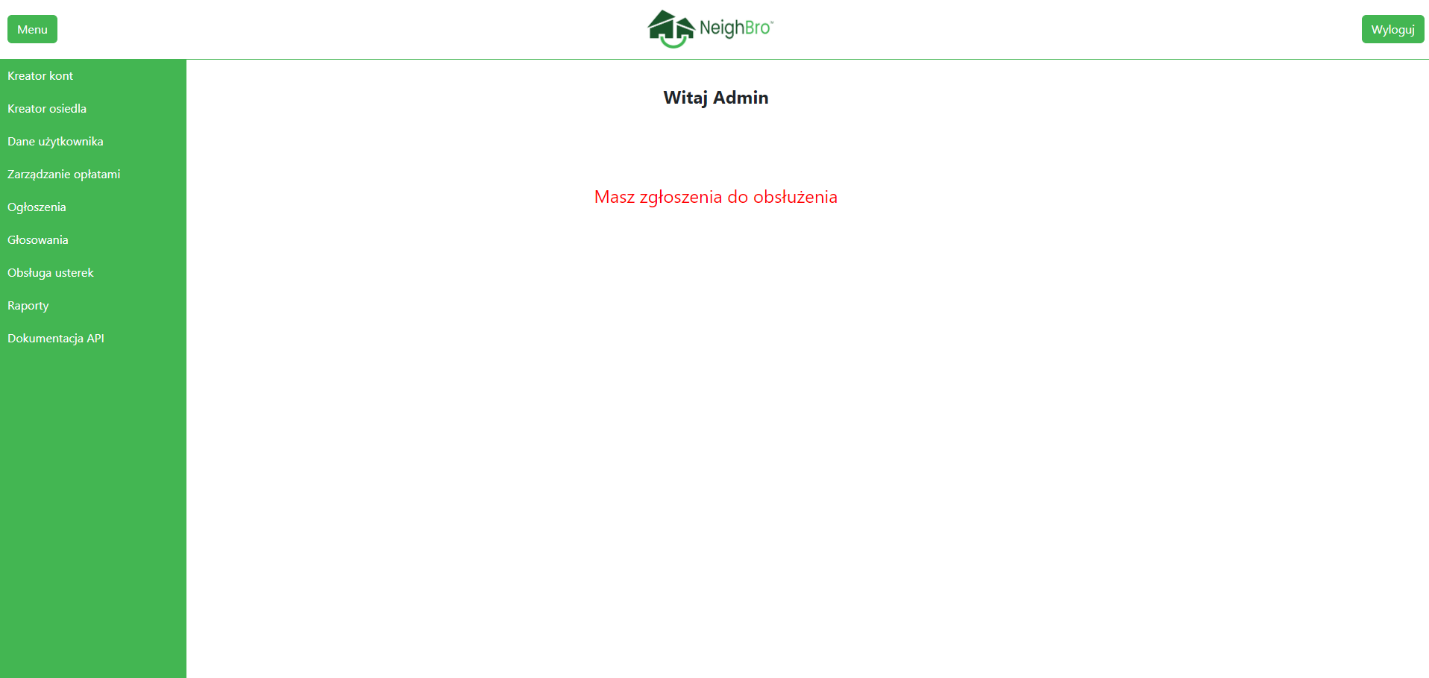
-Powitanie które zawiera nazwę użytkownika podaną podczas tworzenia konta

-Informacje o tym czy znajdują się jakieś zgłoszenia do obsłużenia

Jeśli są jakieś nowe zgłoszenia komunikat zostanie zmieniony.



### Menu

Menu panelu administratora zawiera takie opcje jak:

-Kreator kont służy on do tworzenia nowych kont administratora i użytkownika

-Kreator osiedla służy on do tworzenia budynków, mieszkań, przypisywania właścicieli, wynajmowania powierzchni i wyświetlania tych informacji

-Dane użytkownika opcja ta umożliwia podgląd osobistych danych aktualnie zalogowanego użytkownika, edycje ich oraz zmianę hasła

-Zarządzanie opłatami w tej opcji ustalamy poszczególne opłaty oraz generujemy faktury za nie

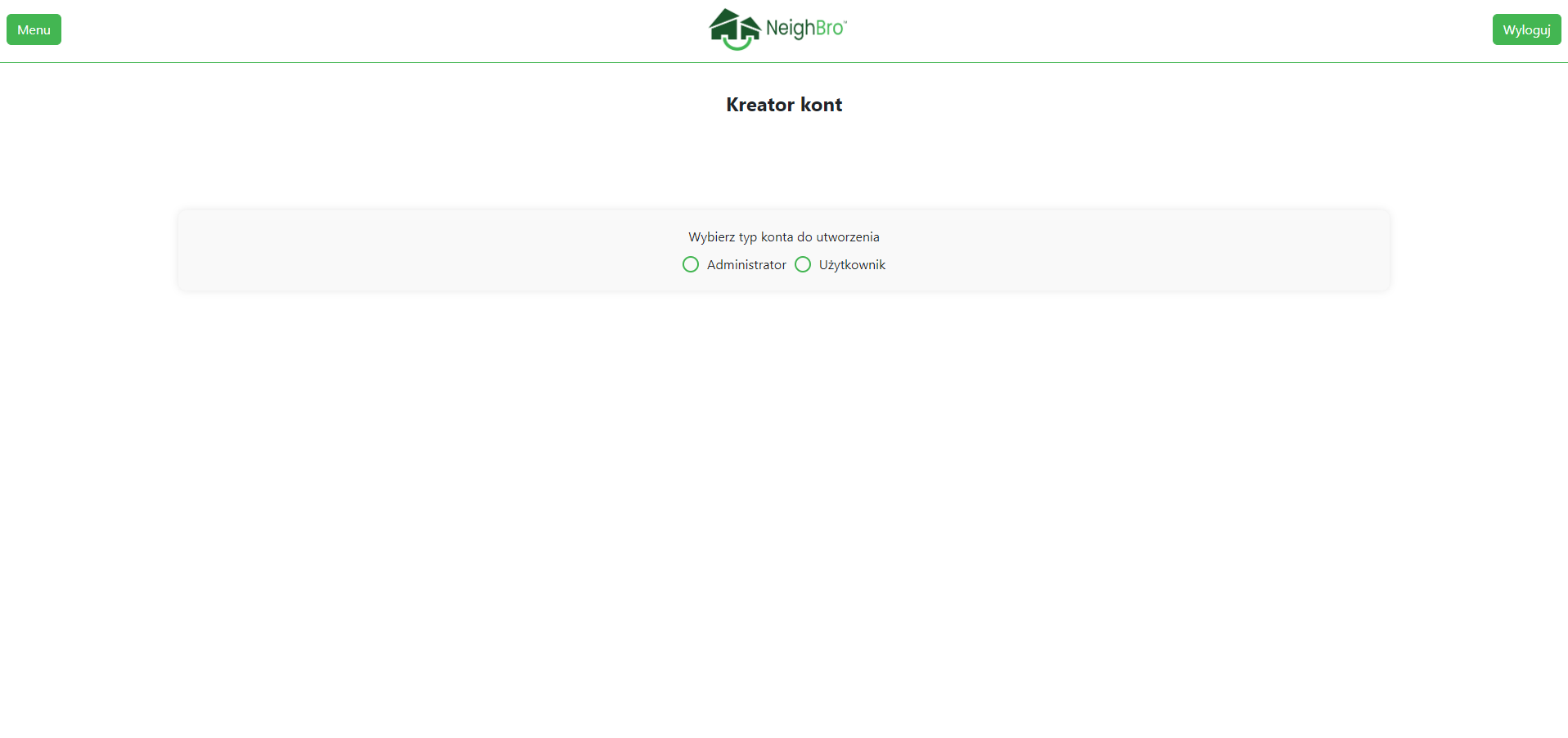
-Ogłoszenia opcja ta pozwala na tworzenie nowych ogłoszeń oraz zarządzanie nimi

-Głosowania w tej pozycji możliwe jest tworzenie nowych głosowań oraz podgląd ich wyników

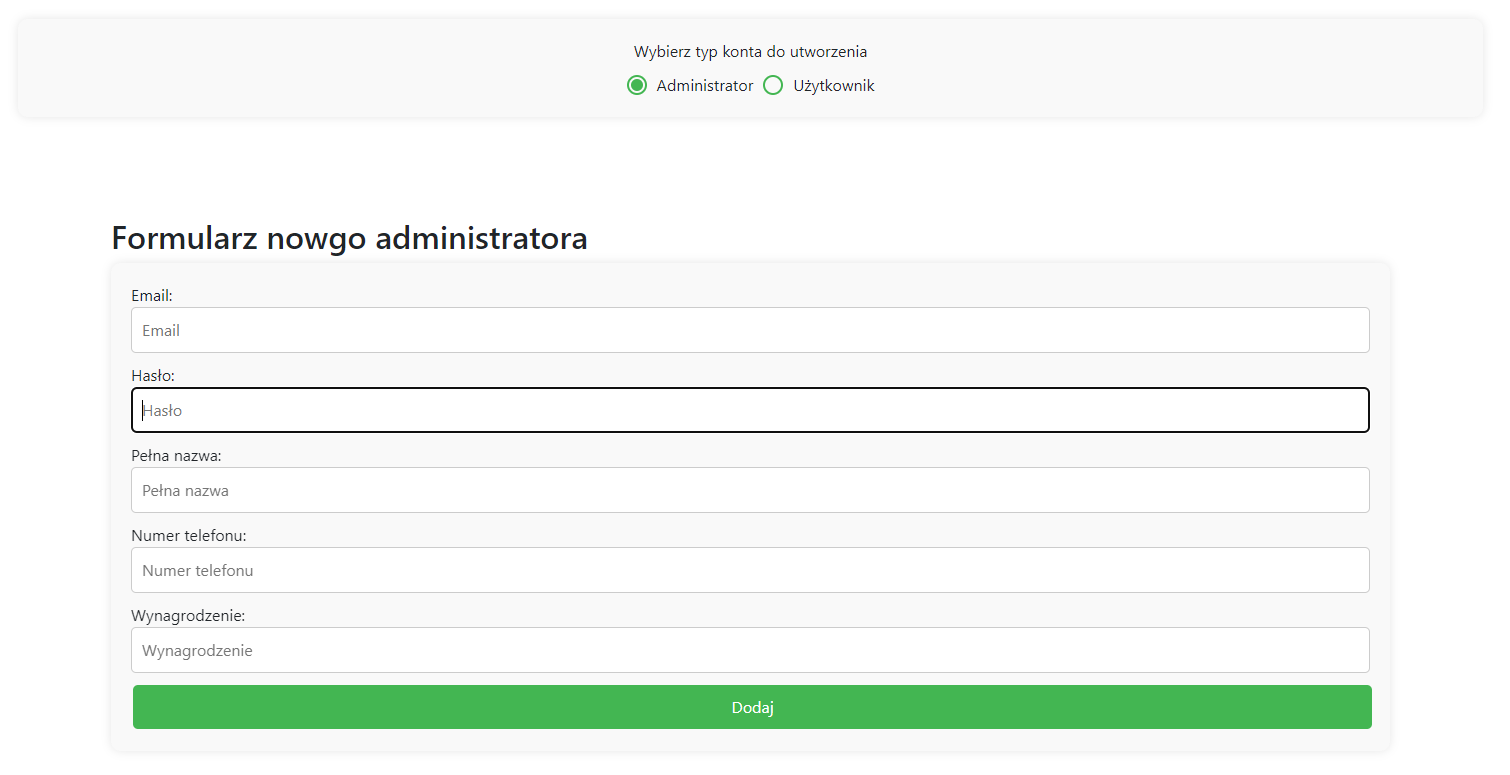
-Raporty tutaj możemy generować raporty PDF

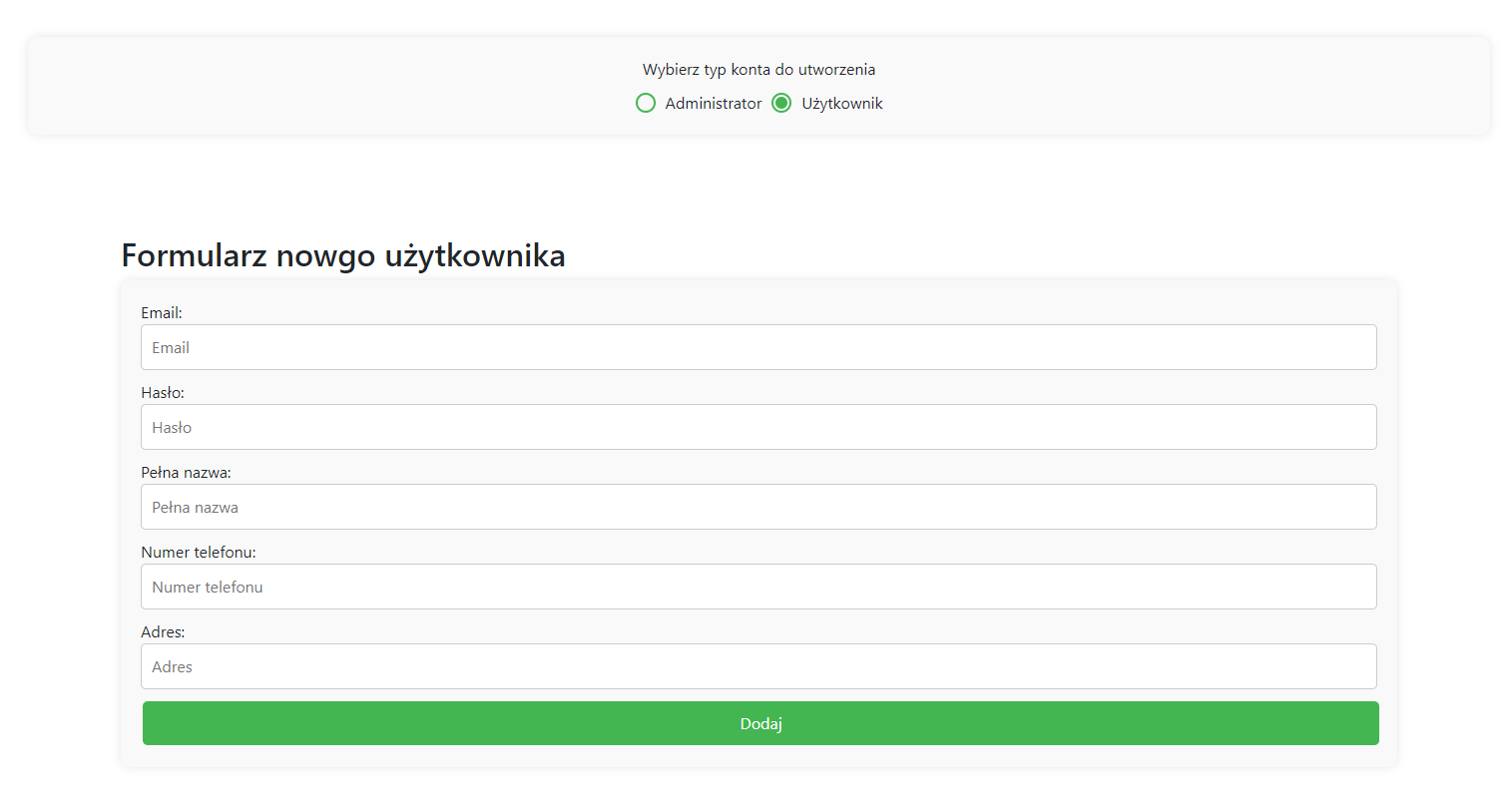
-Dokumentacja API tu możemy zobaczyć jakie requesty może obsłużyć API

### Kreator kont

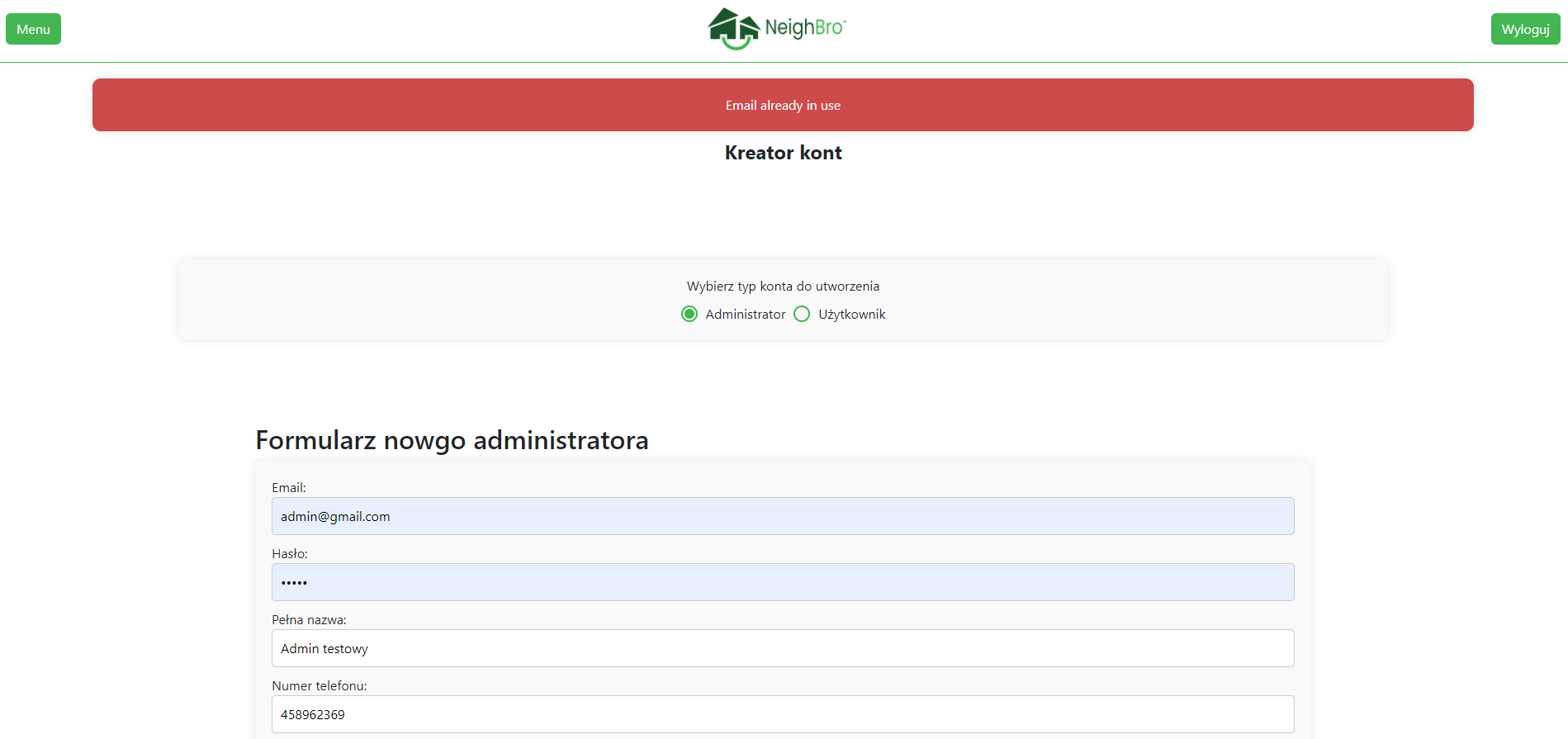


Po wybraniu opcji „Kreator kont” zostanie wyświetlony formularz służący do kreacji nowych kont. Aby utworzyć nowe konto należy wybrać typ konta. Po wybraniu typu zostanie wyświetlony formularz odpowiedni do danego typu konta.

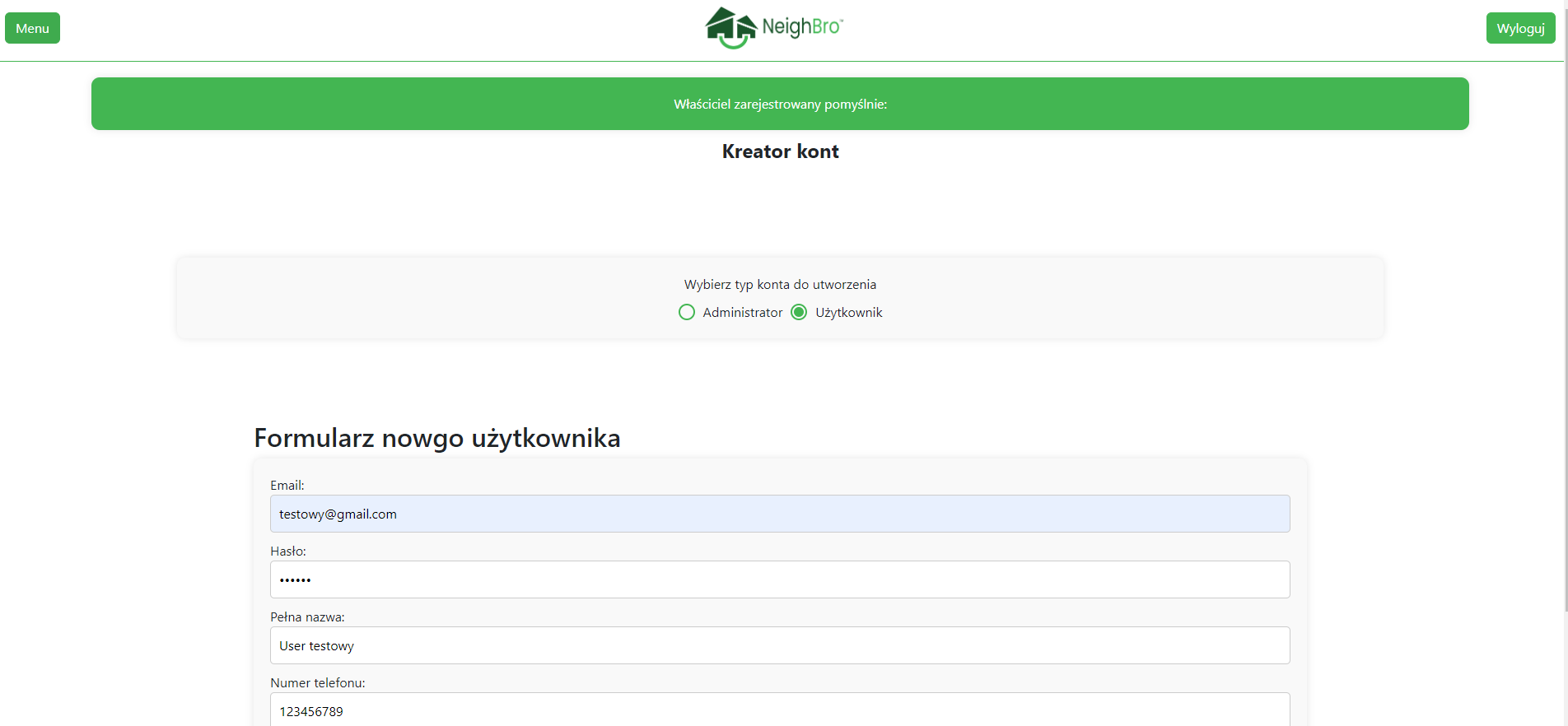




Po wpisaniu danych i wciśnięciu przycisku dodaj zostanie nam wyświetlony odpowiedni komunikat o sukcesie operacji lub jej porażce.



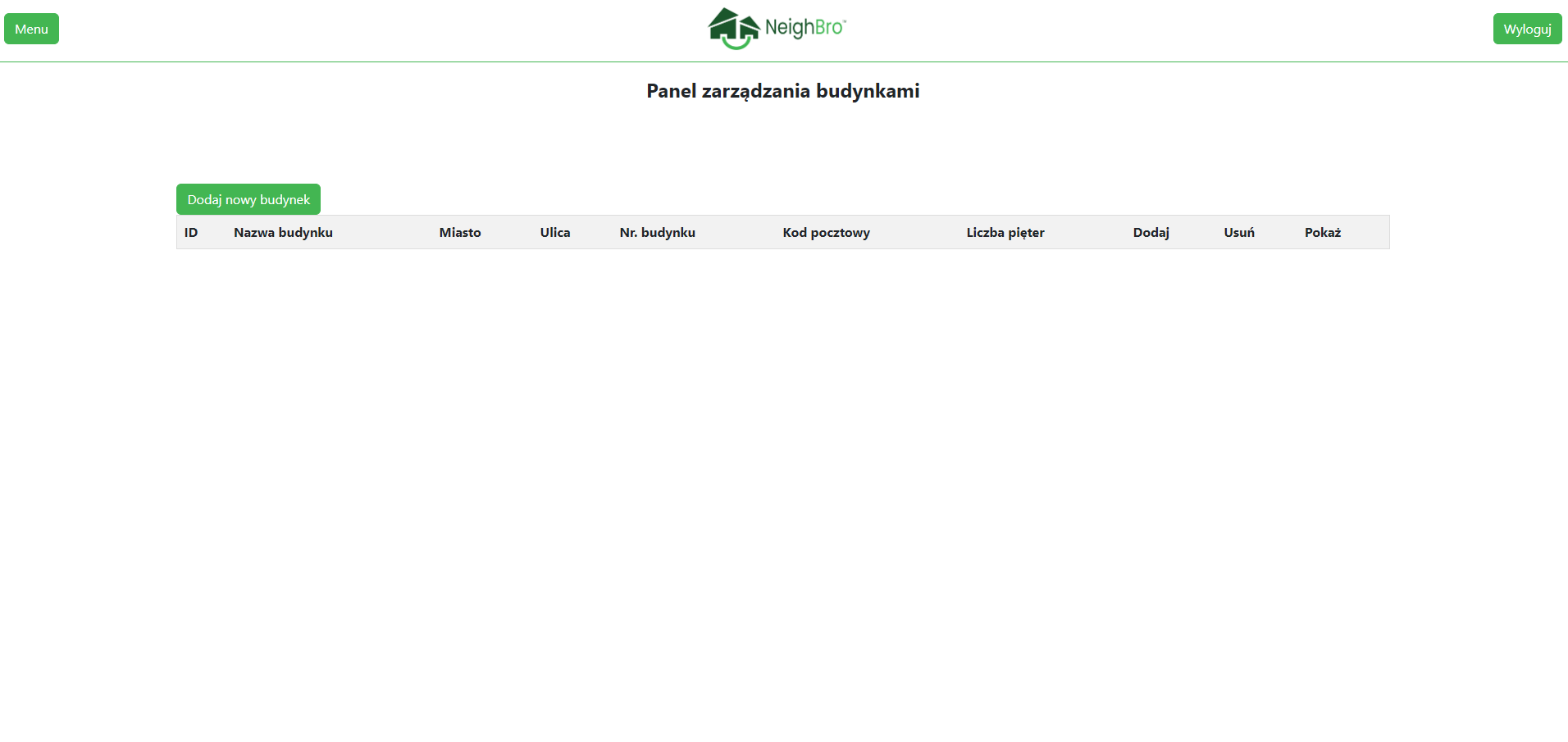
Rysunek 1 Komunikat porażki



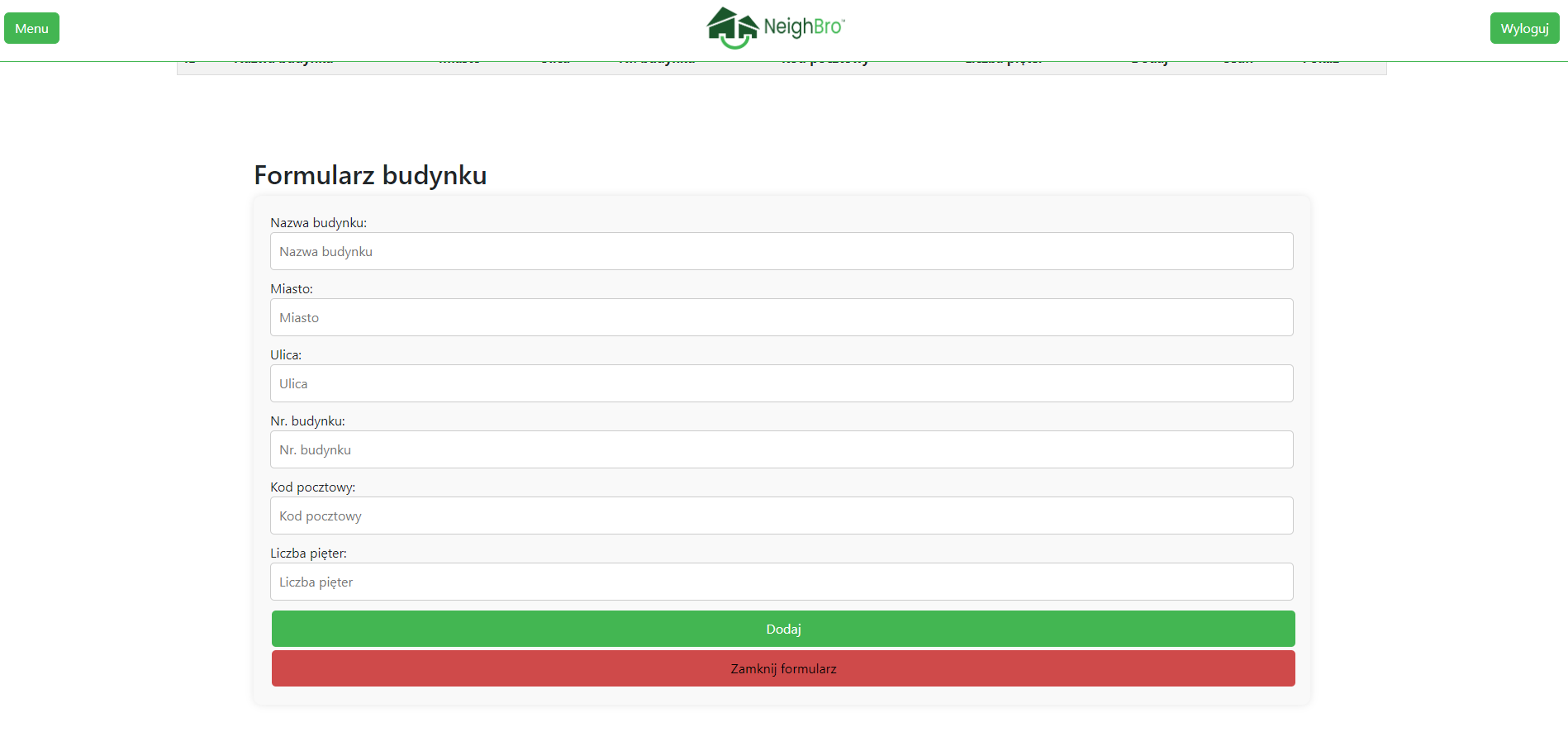
Rysunek 2 Komunikat sukcesu

Komunikat można zamknąć po przez kliknięcie w niego

### Kreator osiedla



Po wybraniu opcji kreator osiedla zostanie wyświetlona tabela z budynkami znajdującymi się w osiedlu. Po kliknięciu w przycisk „Dodaj nowy budynek” zostanie wyświetlony formularz służący do dodania nowego budynku.



Zawiera on pola służące do wpisania informacji o budynku, przycisk „Dodaj” który dodaje budynek do bazy danych oraz przycisk „Zamknij formularz” służący do zamknięcia formularza. Po sukcesie operacji dodawania budynku zostanie on wyświetlony w tabeli. Formularz posiada walidacje na puste pola oraz liczbę pięter która nie może być mniejsza niż 0, reszta walidacji odbywa się na backendzie.

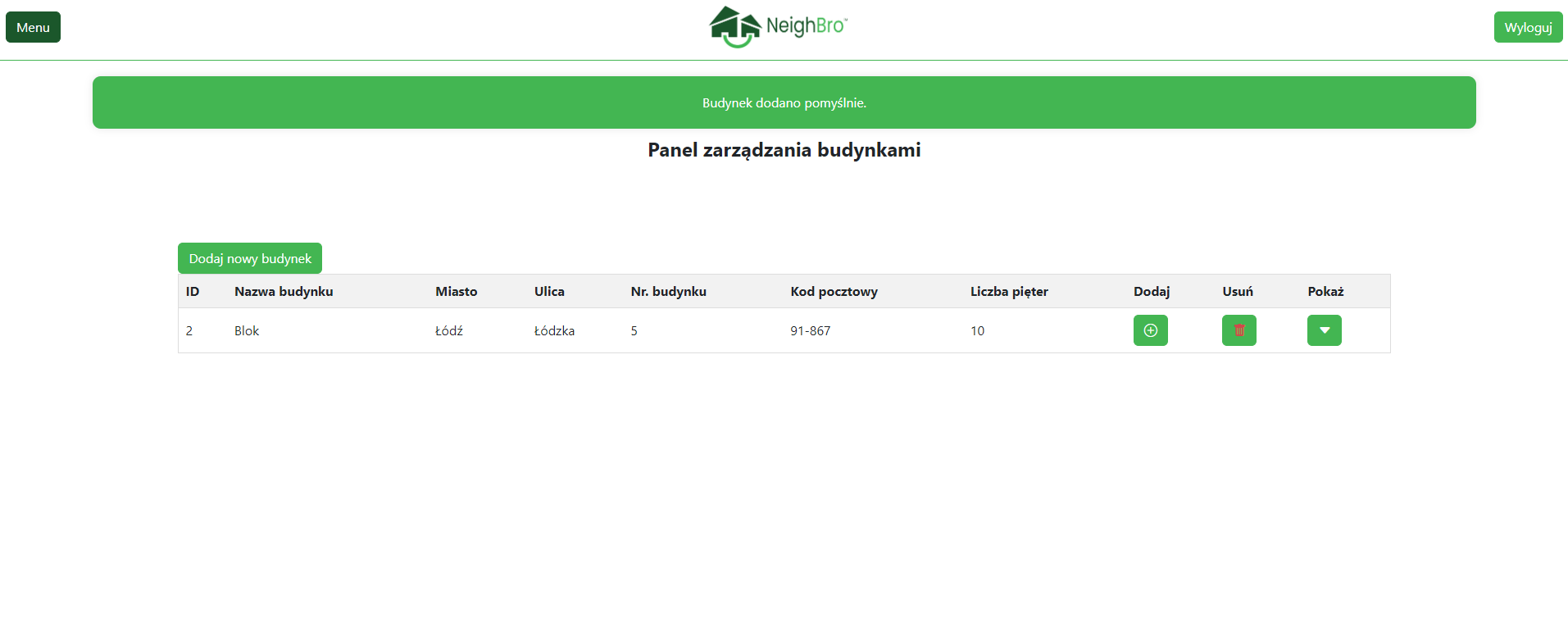


Tabela posiada takie pola jak:

-ID jest to id budynku w bazie danych

-Nazwa budynku nazwa własna budynku

-Miasto ulica nr. budynku kod pocztowy rozbity adres budynku

-Liczba pięter

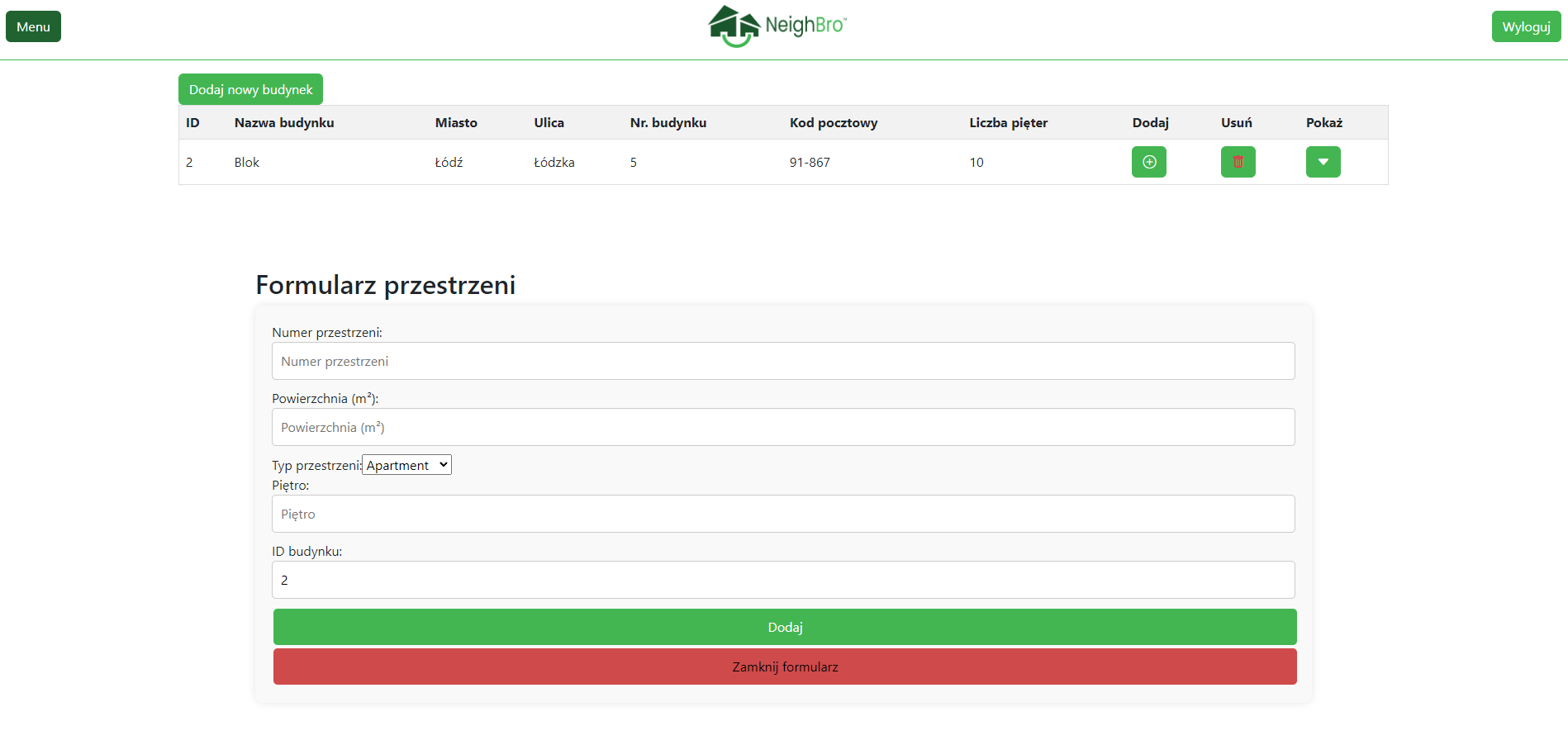
-Opcja „Dodaj” służy ona do dodania do budynku mieszkania

-Opcja „Usuń” służy ona do usuwania budynku

-Opcja „Pokaż” rozwija ona tabele z lista mieszkań w budynku

#### Opcja dodaj

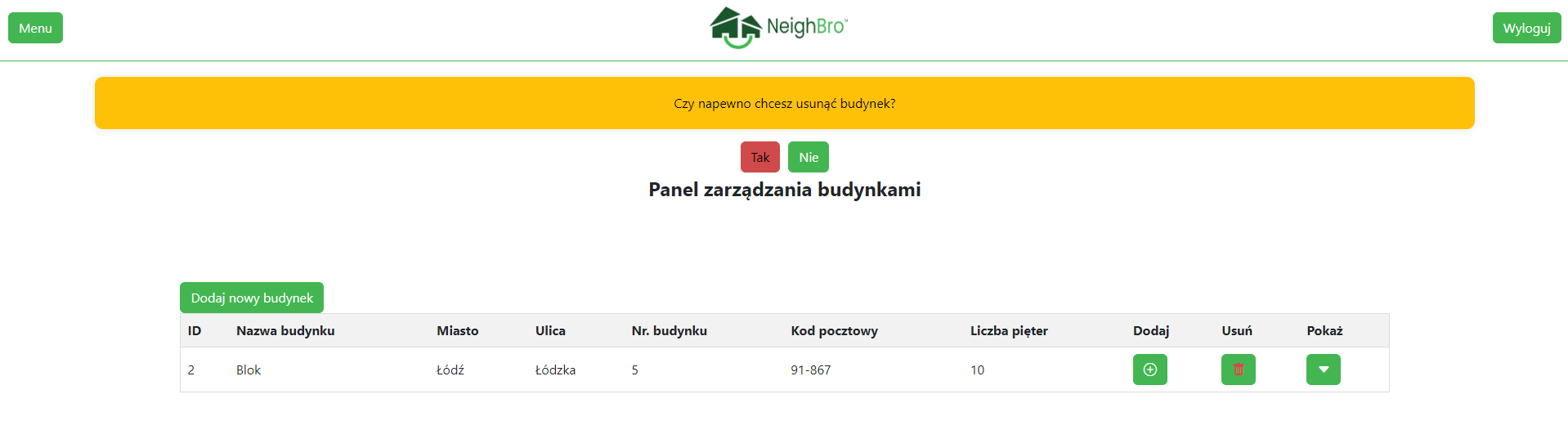
Po wybraniu tej opcji zostanie wyświetlony formularz dodania mieszkania do budynku.

Lista typów przestrzeni jest generowana na podstawie listy z bazy danych a id budynku pobierane z wiersza w którym została wybrana opcja „Dodaj” to pole jest zablokowane na zmiany wprowadzane przez użytkownika.

Formularz posiada walidacje na pola puste, powierzchnie która nie może być mniejsza od 0, na piętro które nie może być mniejsza od 0 reszta walidacji odbywa się na backendzie. Po sukcesie operacji dodawania budynku zostanie on wyświetlony w tabeli.

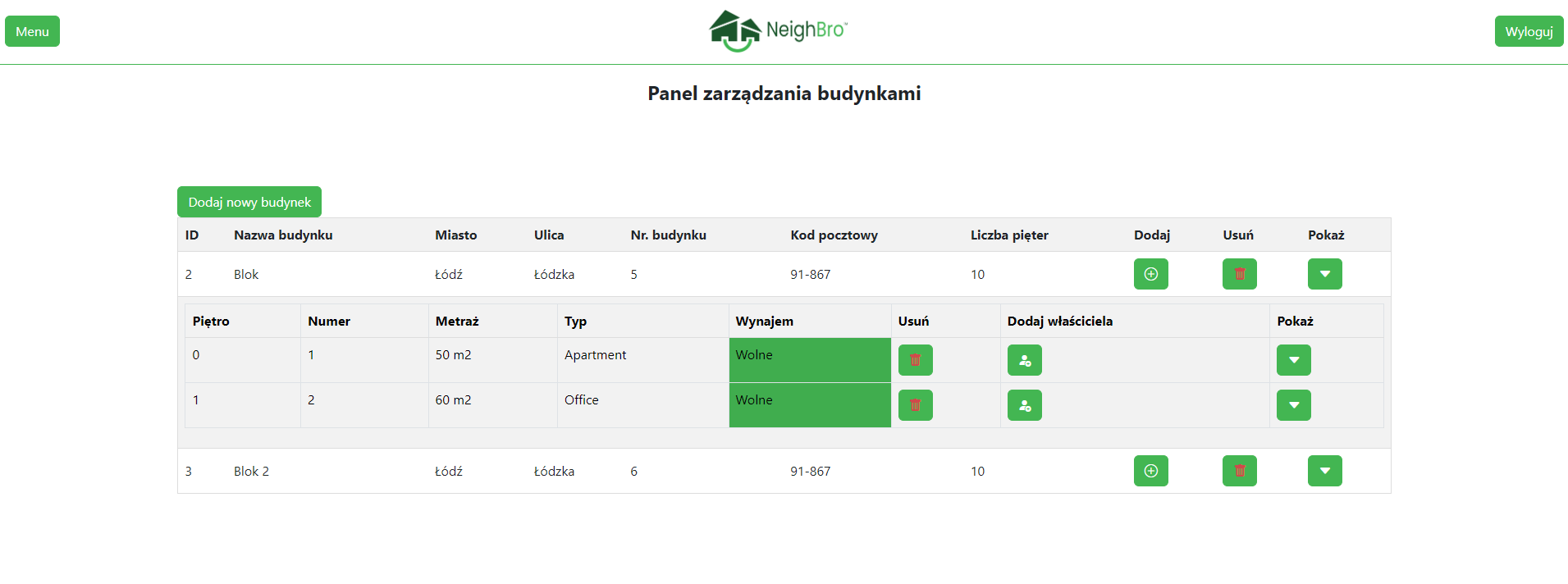
#### Opcja „Usuń”

Służy ona do usuwania budynku z bazy danych przed usunięciem zostanie wyświetlony komunikat proszący o potwierdzenie operacji.



#### Opcja „Pokaż”

Służy ona do rozwinięcia tabeli z listą mieszkań w danym budynku.



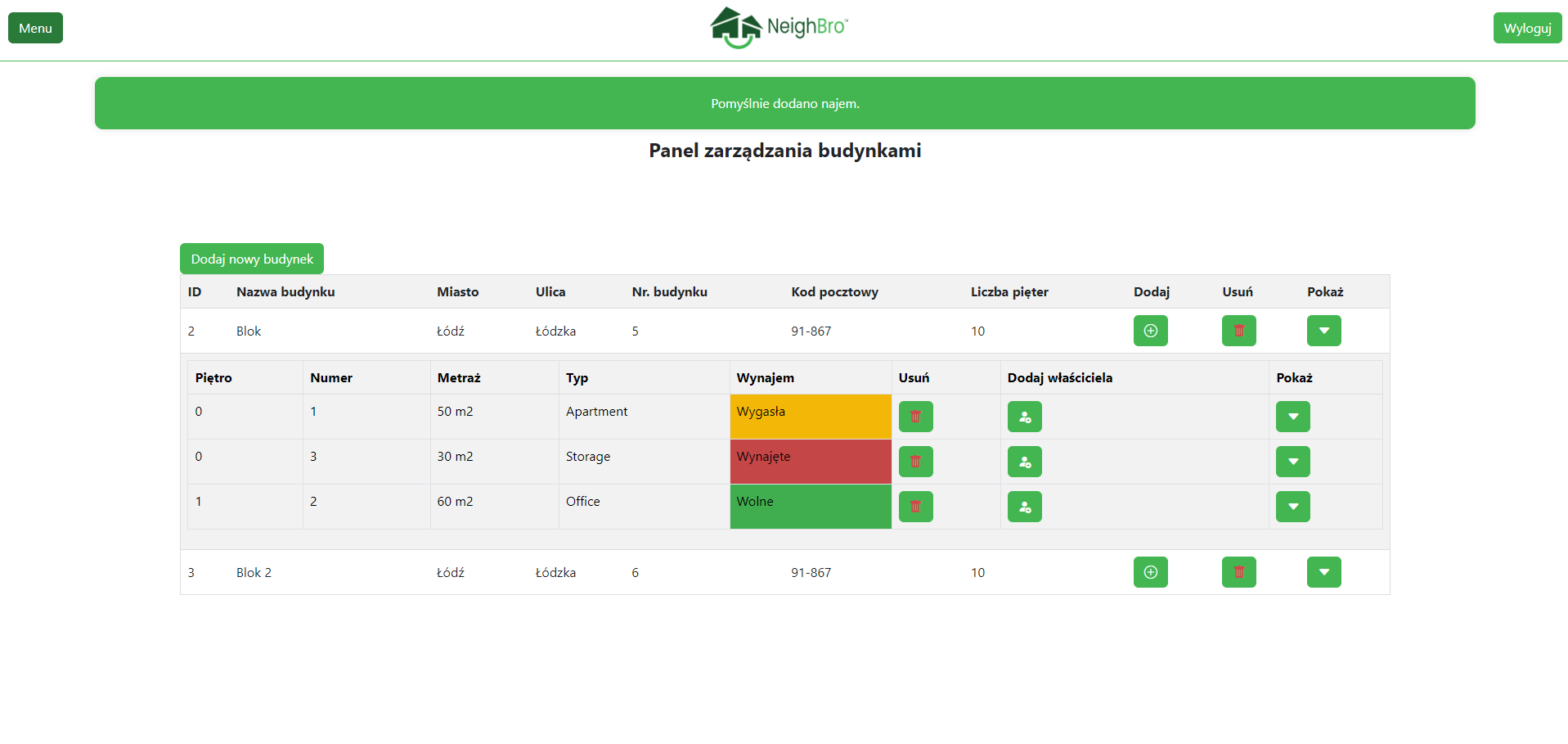
W tej tabeli możemy mamy możliwość:

-Wynajęcia przestrzeni robimy to poprzez klikniecie pola w kolumnie wynajem

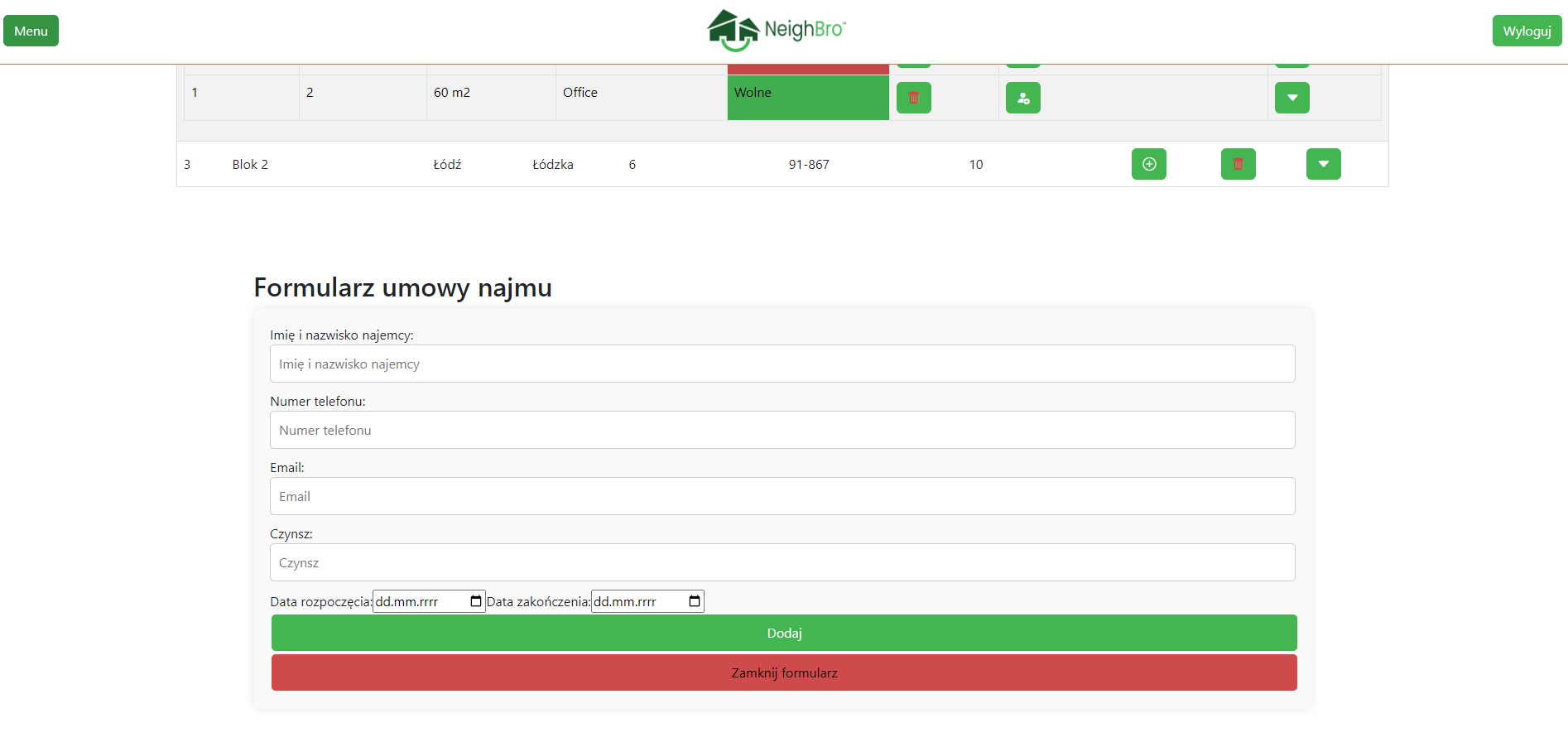
-Usunięcie przestrzeni robimy to poprzez klikniecie przycisku w kolumnie usuń działa tak samo jak w przypadku usuwania budynku

-Pokaż służy do wyświetlania szczegółowych informacji o przestrzeni takich jak tabela właścicieli, mieszkańców i umowy wynajmu

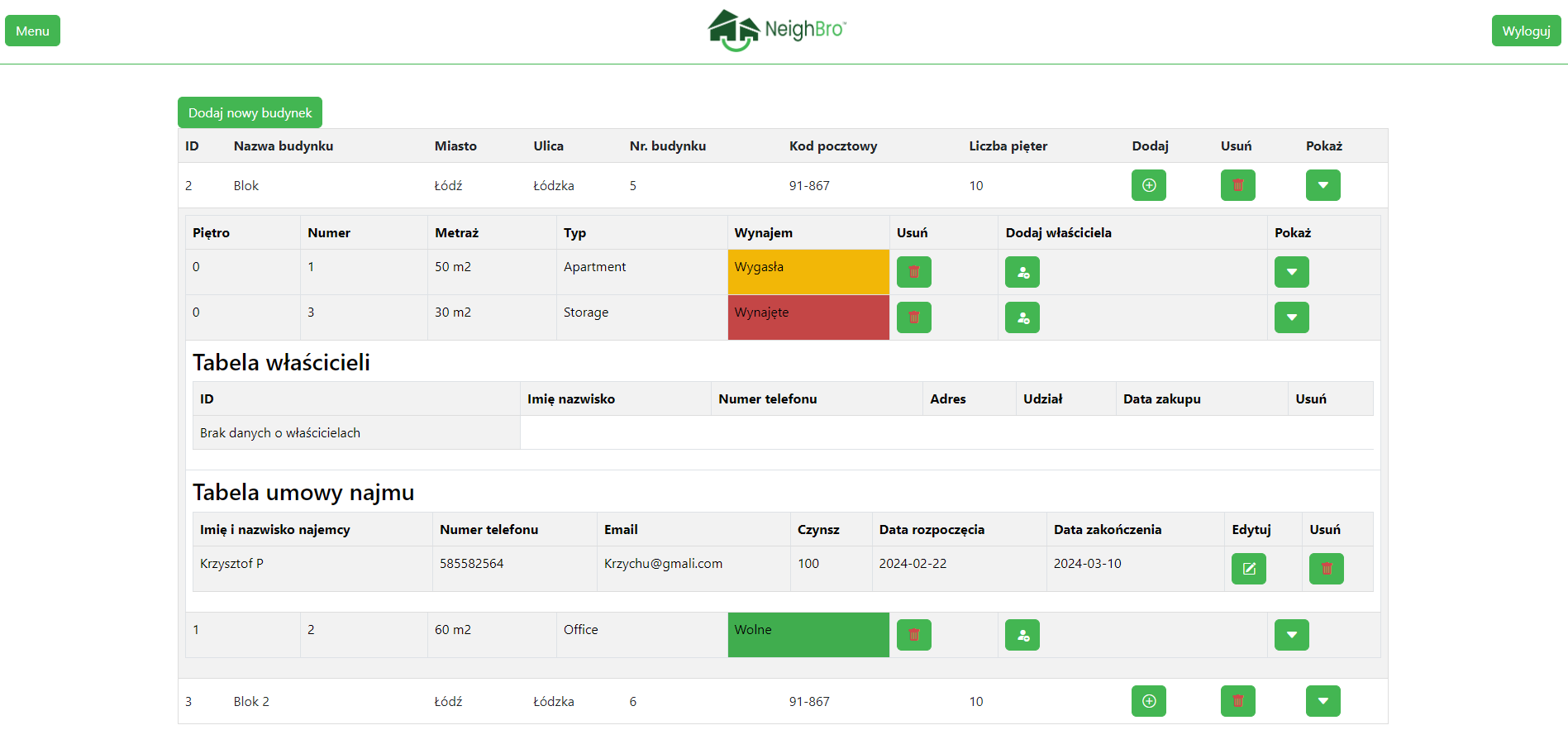
#### Wynajem powierzchni



Wynajem odbywa się po przez kliknięcie w pole w kolumnie wynajem. Możliwe jest wynajęcie przestrzeni która ma status „Wolne”. Po kliknięciu zostanie wyświetlony formularz umowy wynajmu.



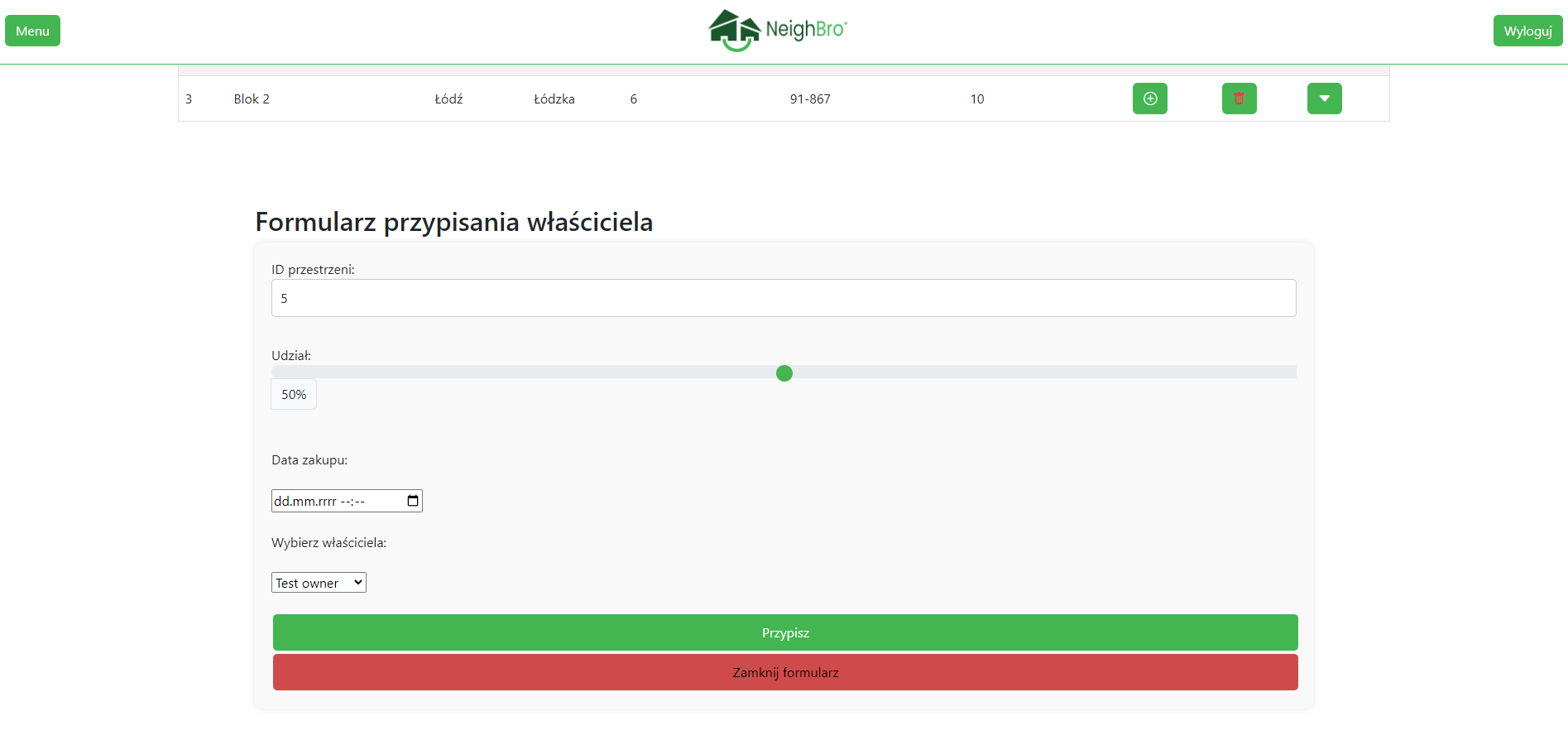
W tym formularzu wpisujemy dane najemcy oraz okres w jakim umowa będzie obowiązywać. Po sukcesie operacji wynajmu umowa pokaże się w tabeli z szczegółami powierzchni. Walidacja jest na pola puste, pole czynsz, który nie może być mniejszy od 0 oraz na format email reszta walidacji odbywa się na backendzie.



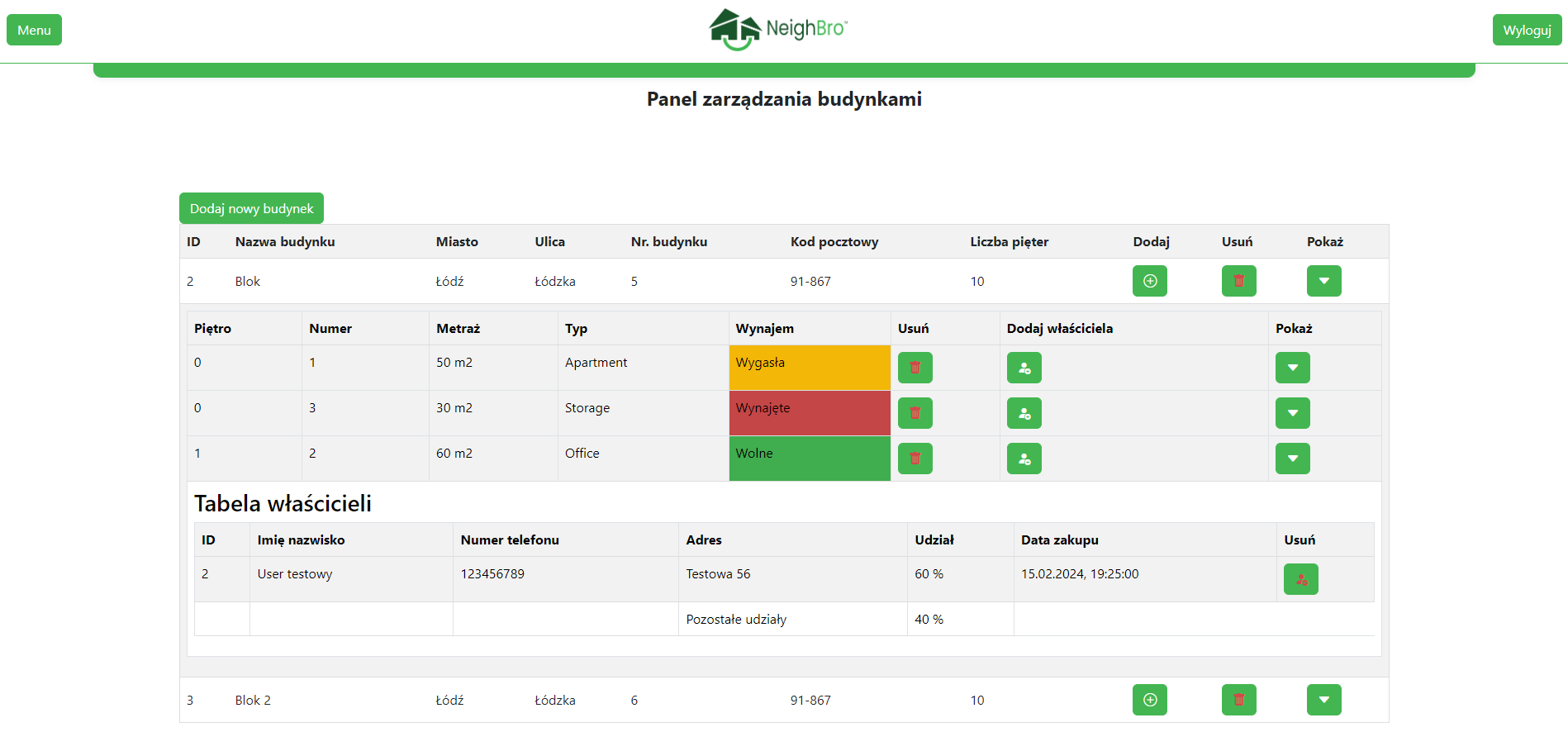
Tutaj też możemy ją edytować lub usunąć.

#### Dodaj właściciela

Po wybraniu tej opcji zostanie ukazany formularz dodawania właściciela.

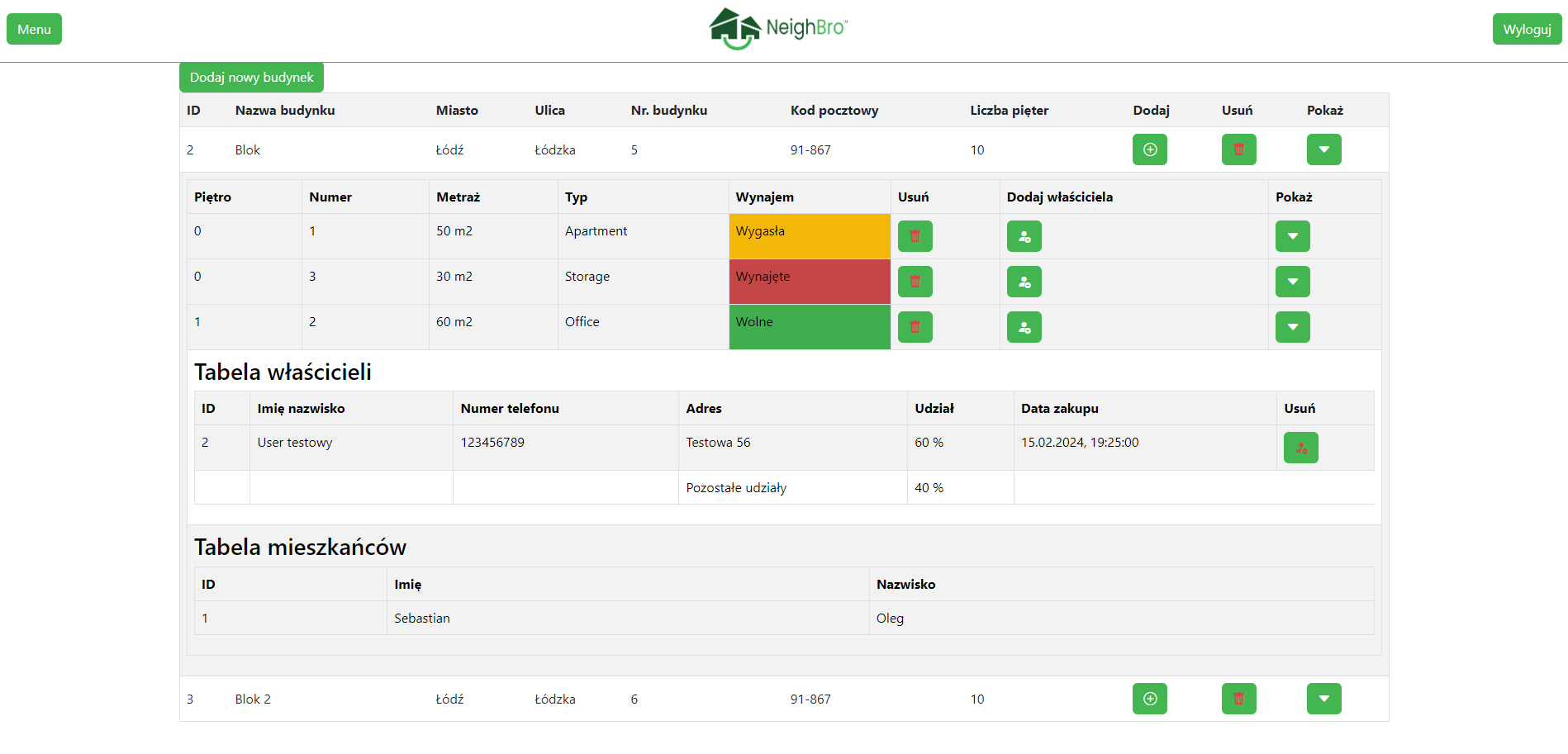


W formularzu wpisujemy jaki udział w mieszkaniu ma właściciele kiedy dokonano zakupu oraz kto jest właścicielem. Walidacja jest na pola puste reszta walidacji odbywa się na backendzie. Po dodani właściciela jego dane zostaną pokazane w tabeli.

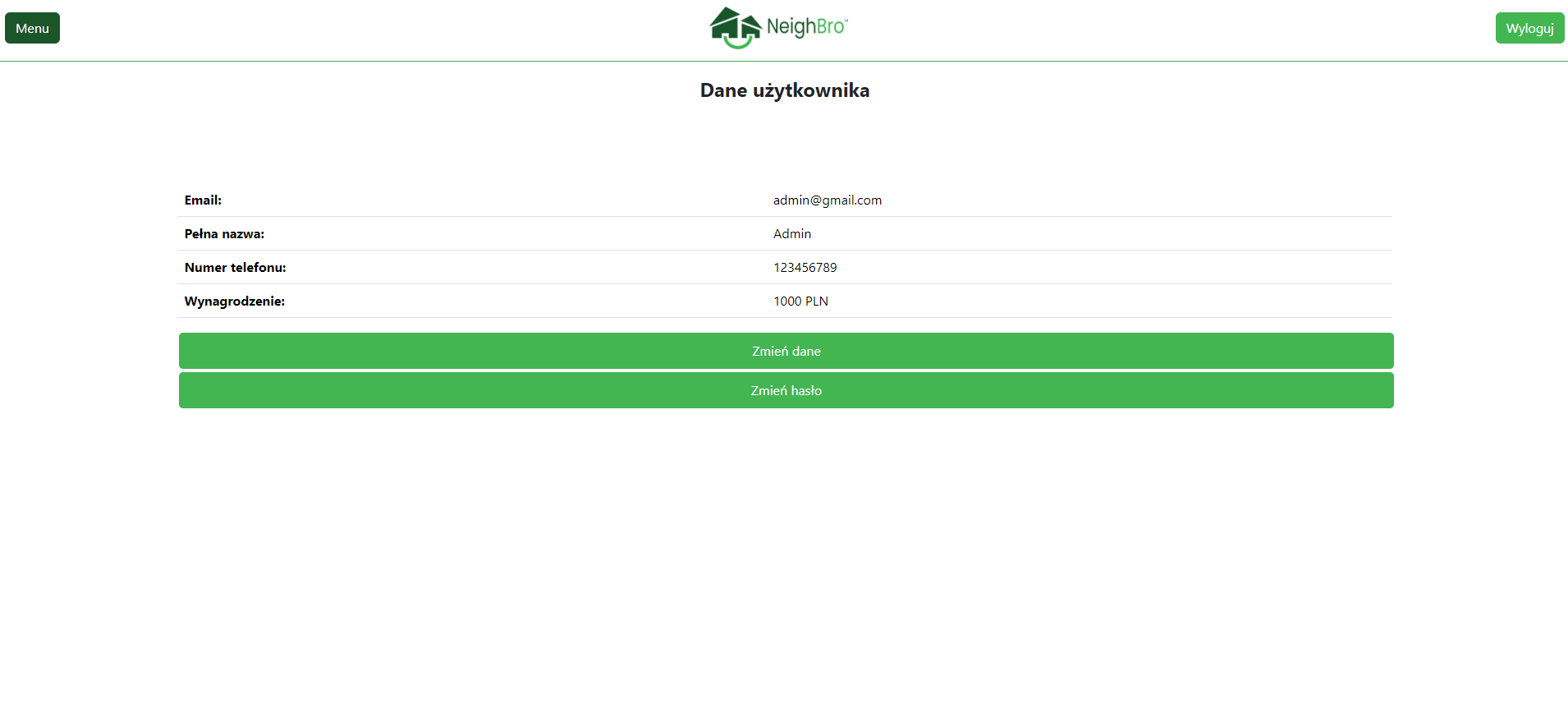


#### Pokaż

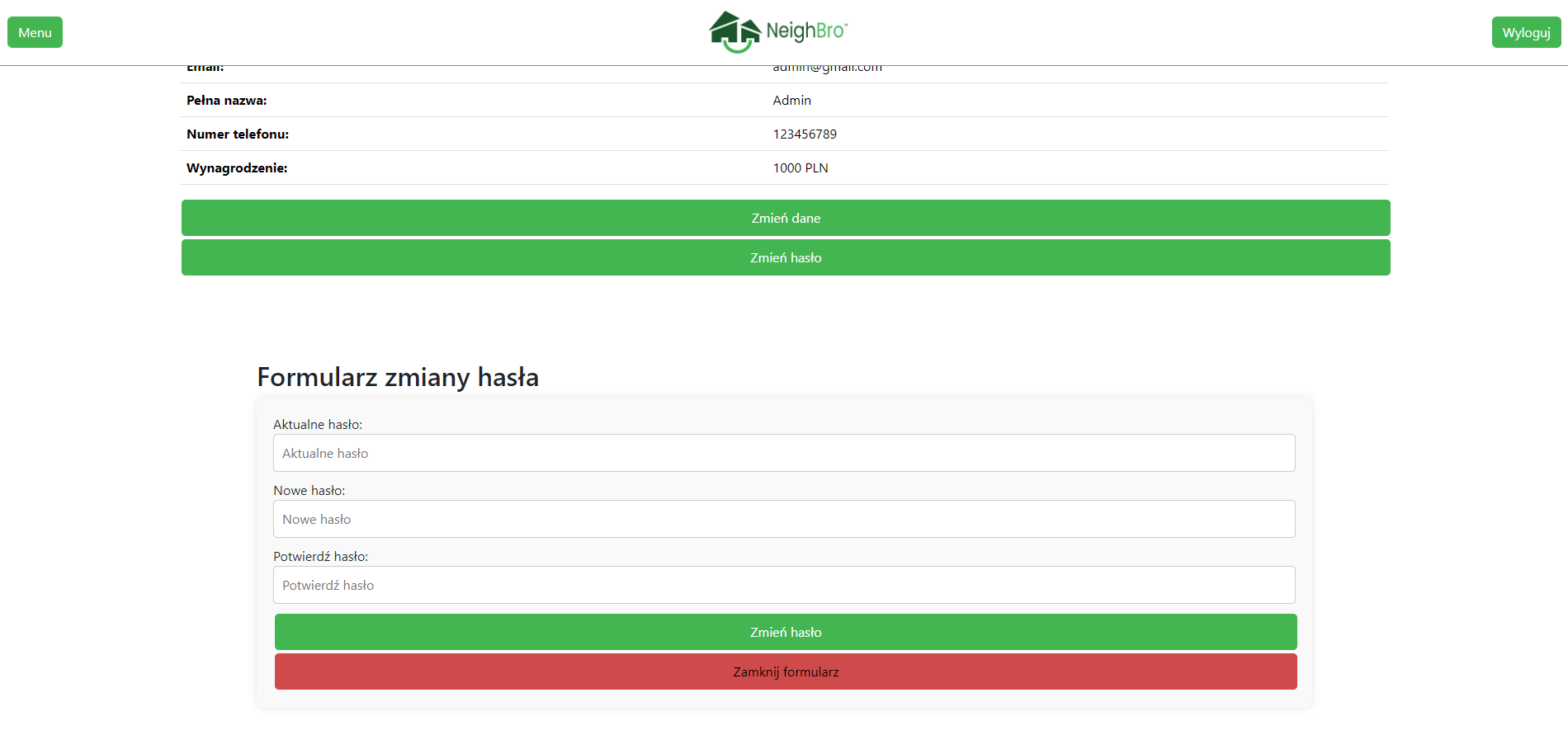
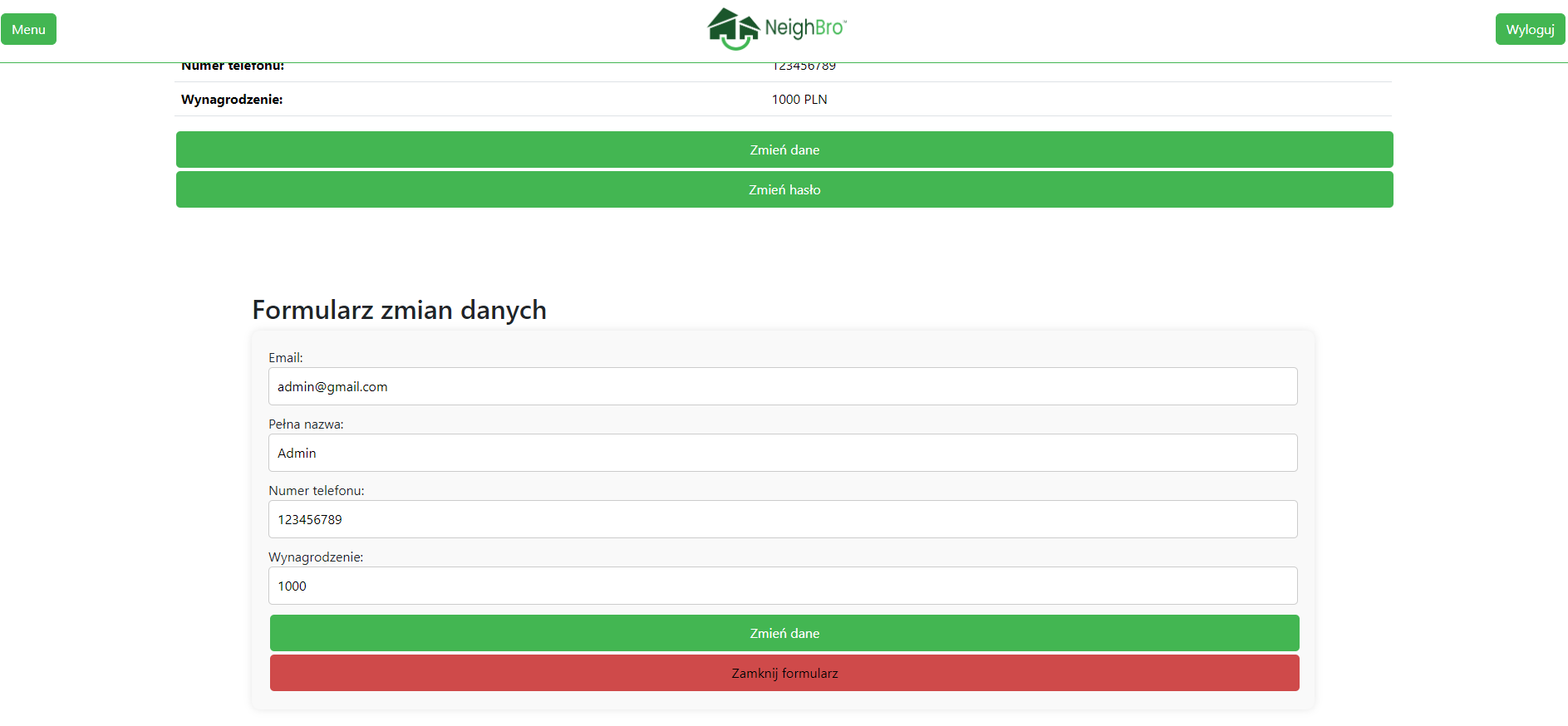
Opcja ta rozwija szczegóły powierzchni takie jak tabela wynajmu jeżeli powierzchnia jest wynajęta, tabela właścicieli oraz tabela mieszkańców.



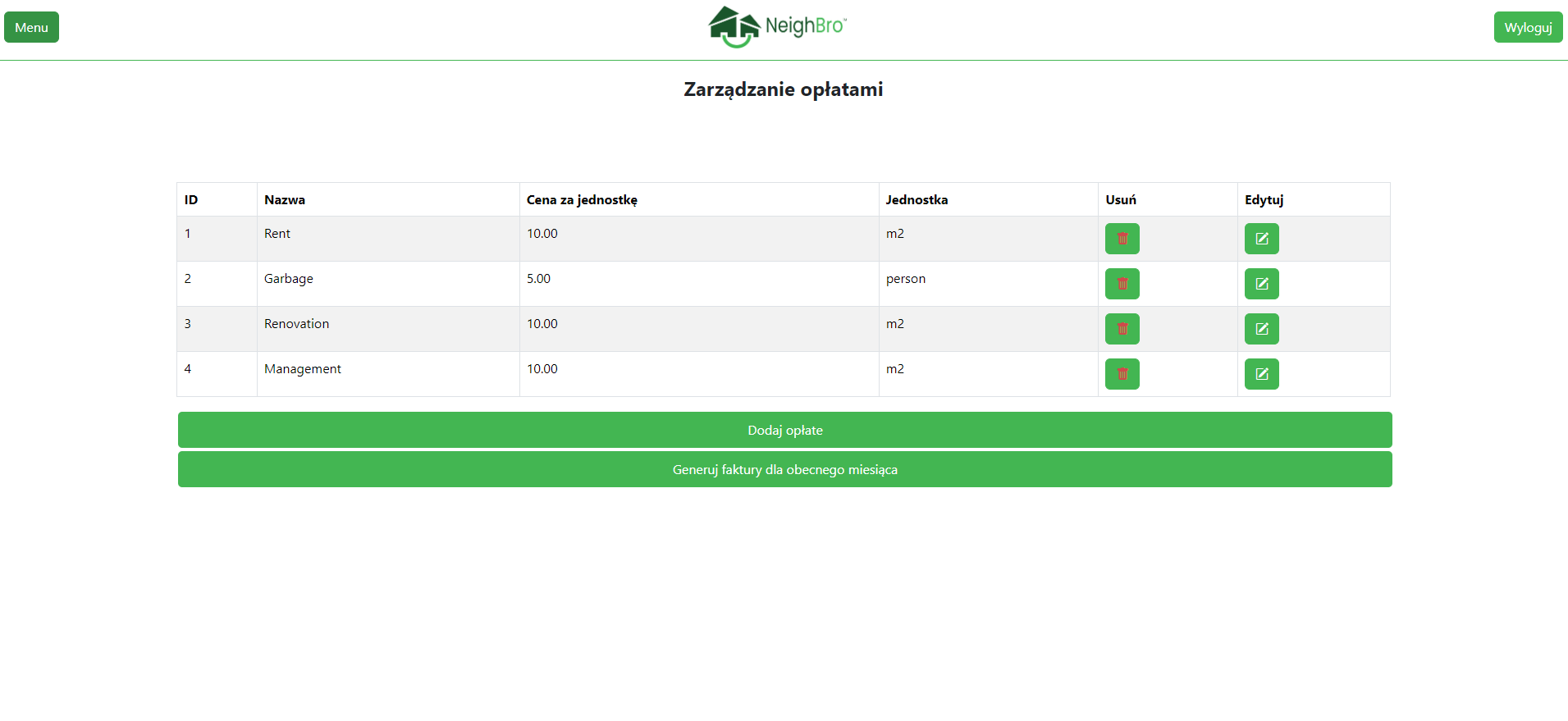
### Dane użytkownika



Po wybraniu tej opcji użytkownikowi zostaną ukazane jego dane oraz dwa przyciski umożliwiające zmianę danych oraz hasła.

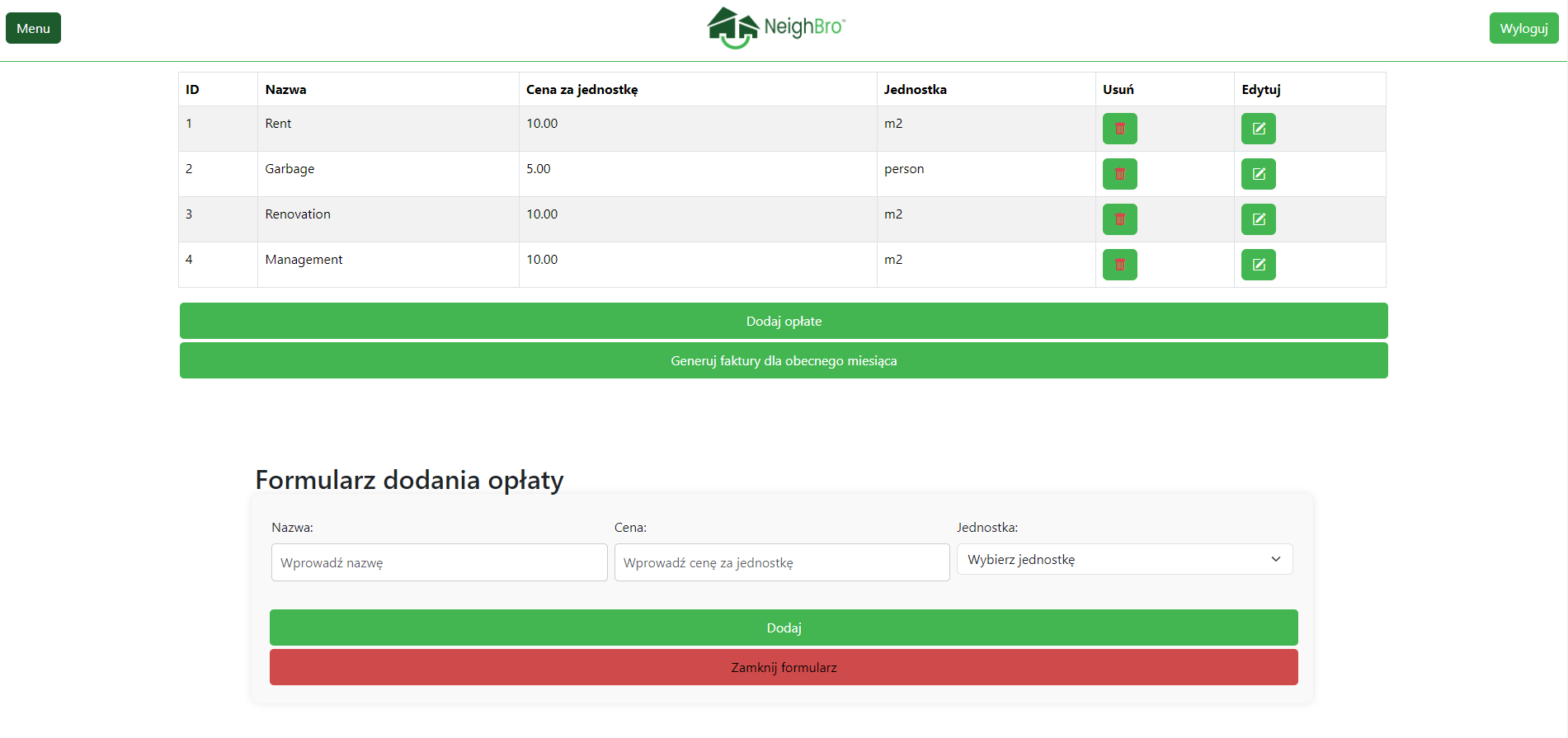


### Zarządzanie opłatami



Opcja ta służy do edycji opłat oraz generowani miesięcznych faktur za nie. Po wciśnięciu przycisku „Dodaj opłatę” ukaże się formularz dodania opłaty.

#### Dodaj opłate

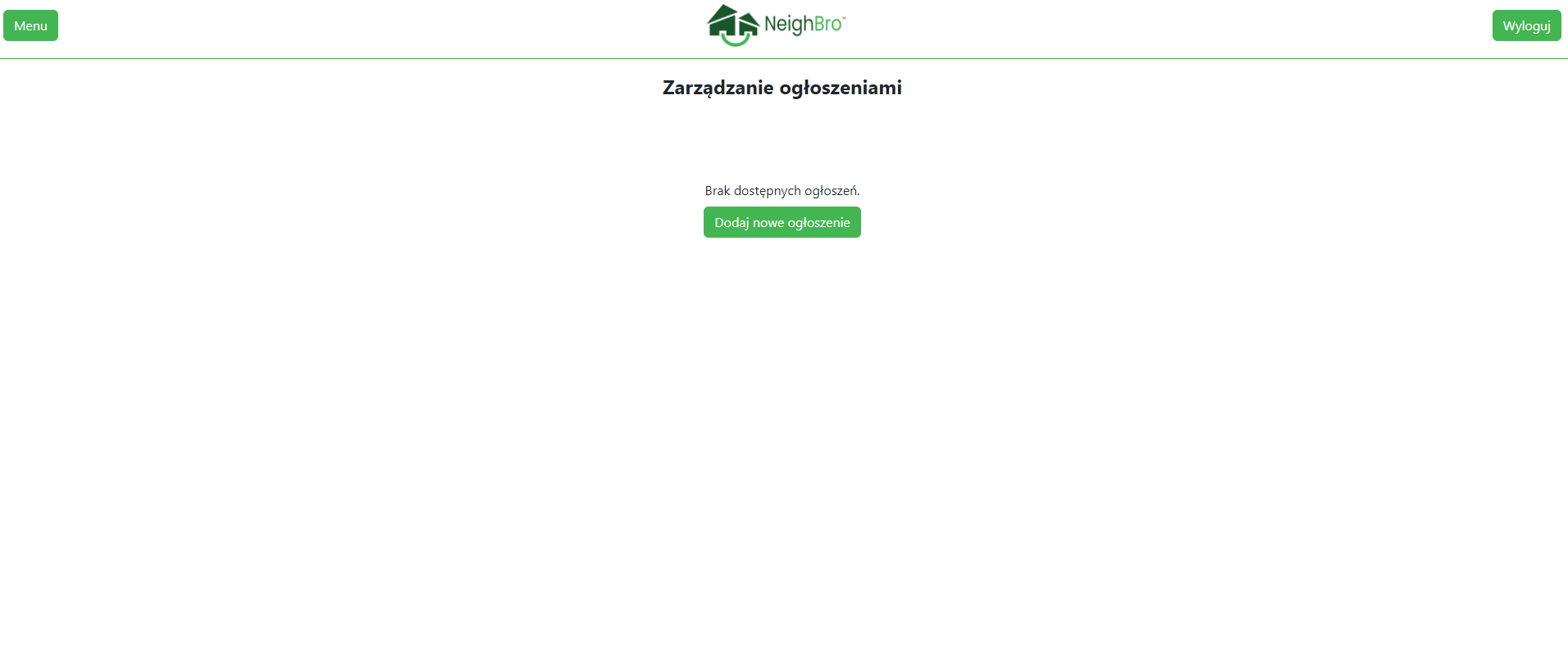


W formularzu wpisujemy nazwę opłaty jej cenę za jednostek oraz podstawę rozliczenia (za osobę lub za m2 powierzchni). Walidacja jest na pola puste oraz na cenę która nie może być mniejsza bądź równa 0.

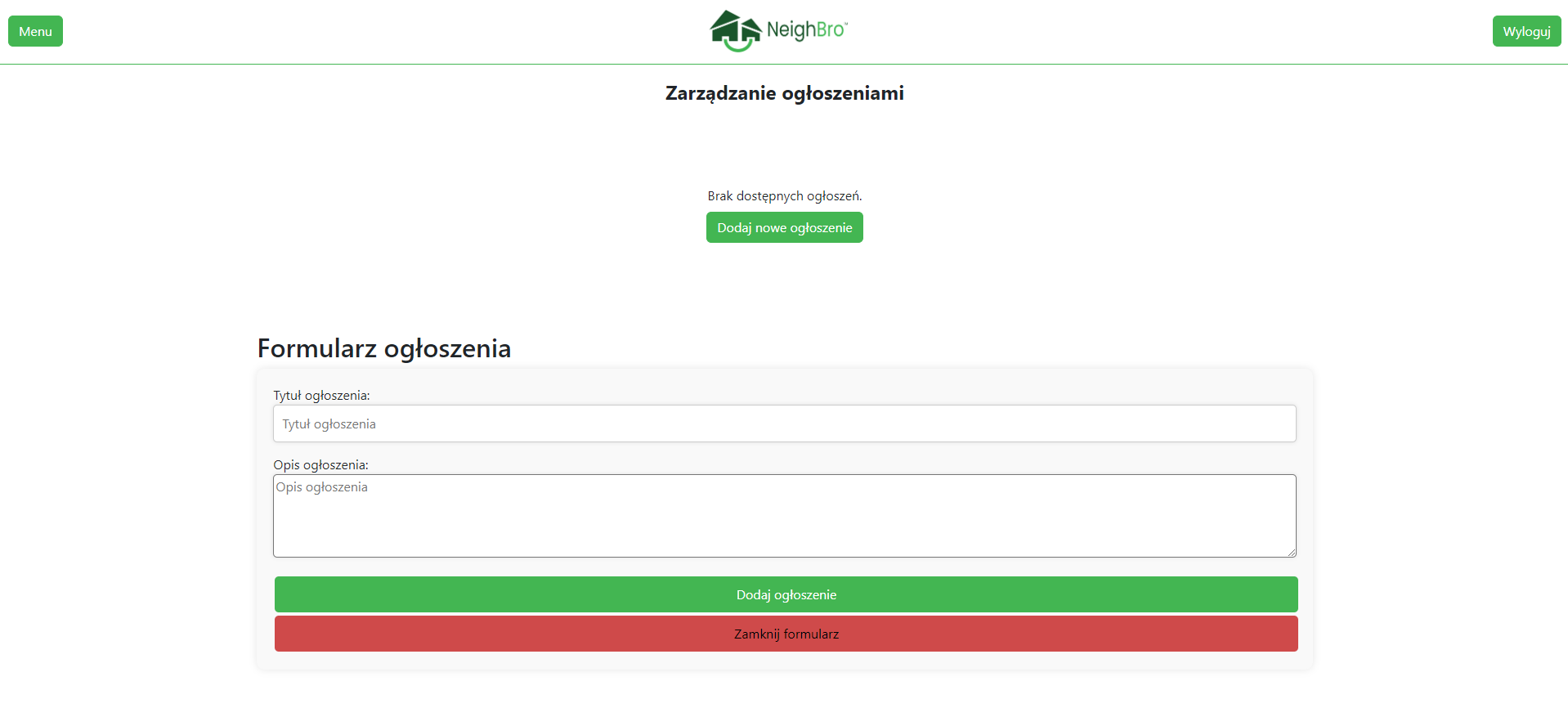
#### Generuj faktury dla obecnego miesiąca

Przycisk ten służy do generowania faktur dla całego osiedla za aktualny miesiąc jeżeli faktury zostały już wygenerowane zostanie wyświetlony odpowiedni komunikat.

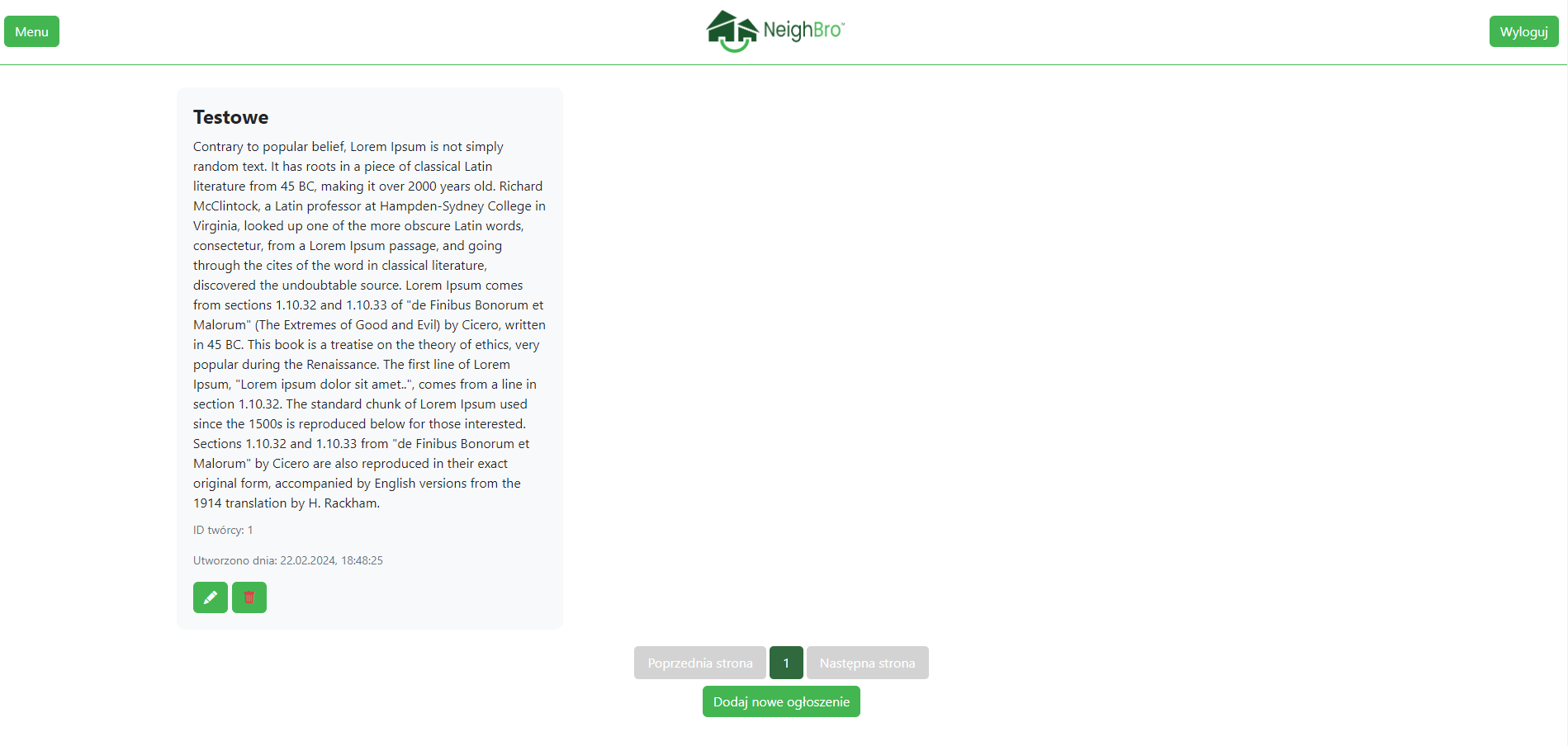
### Ogłoszenia



W tej pozycji w menu możemy dodawać nowe ogłoszenia które będą widoczne dla właścicieli. Aby stworzyć nowe ogłoszenie należy wcisnąć przycisk „Dodaj nowe ogłoszenie”. Po jego wciśnięciu zostanie wyświetlony formularz.



Walidacja na pola puste.

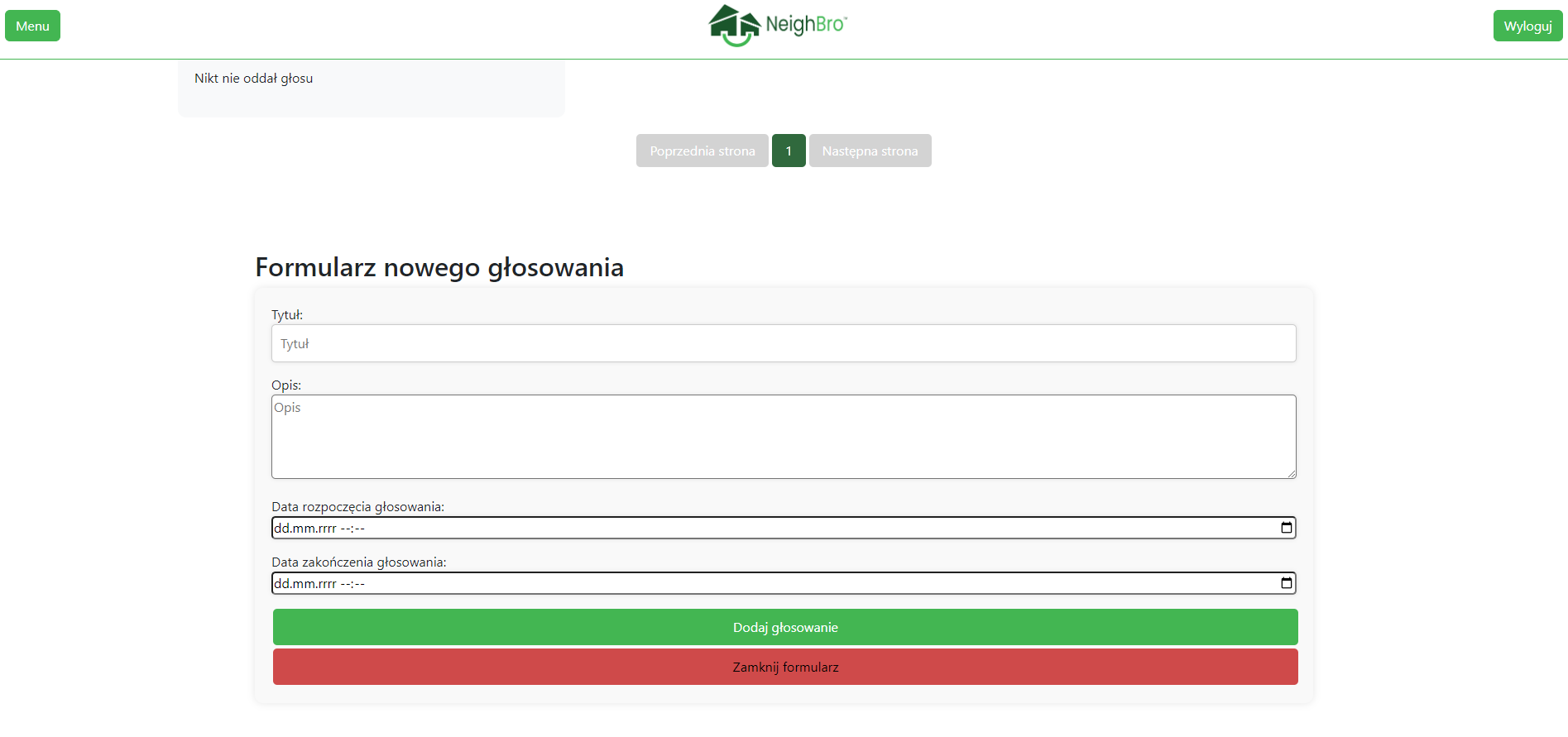


Po dodaniu ogłoszenia możemy je edytować lub usunąć. Ponadto ogłoszenia posiadają paginacje.

### Głosowania

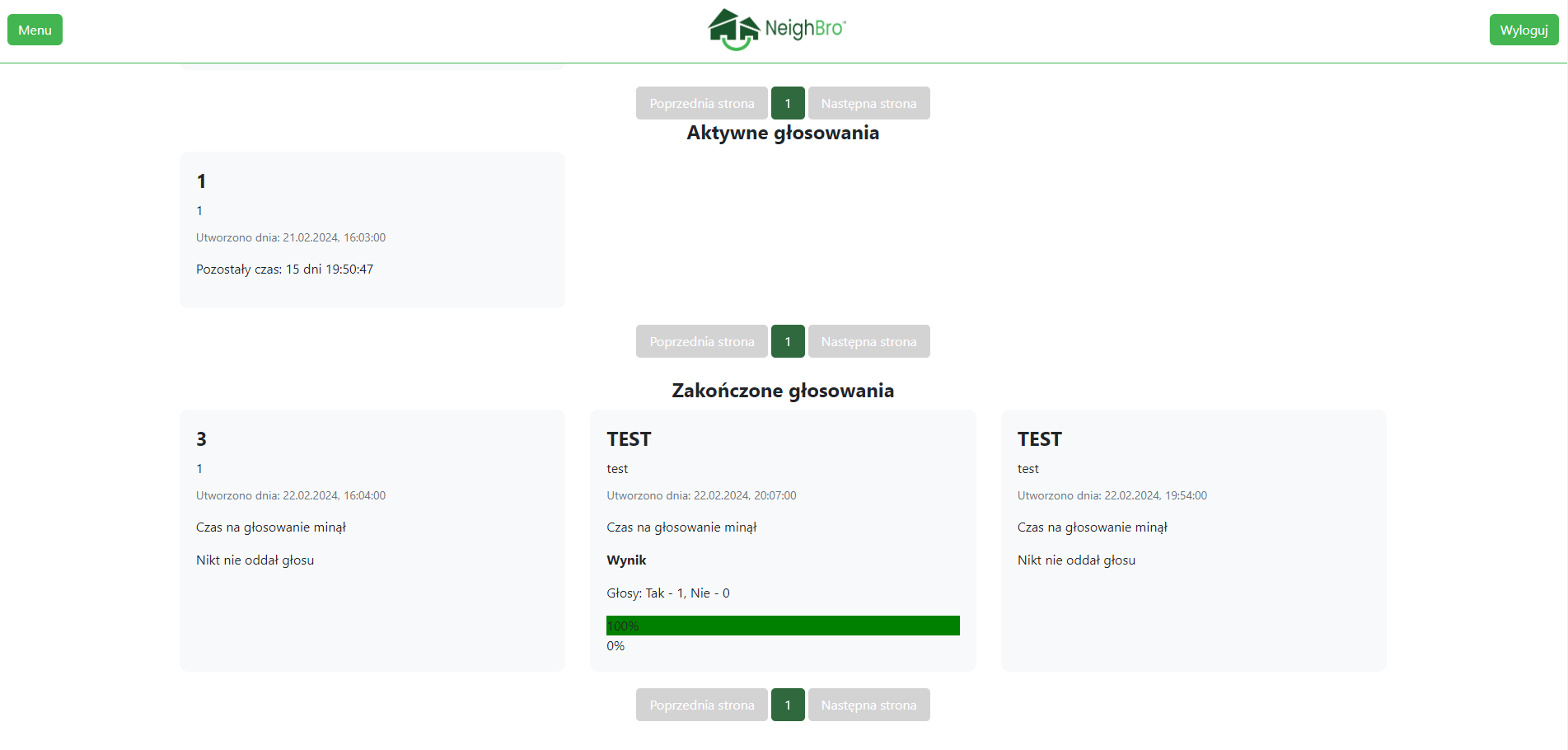


Opcja ta służy do tworzenia nowych glosowań. Aby utworzyć nowe ogłoszenie należy wcisnąć przycisk „Dodaj głosowanie”.

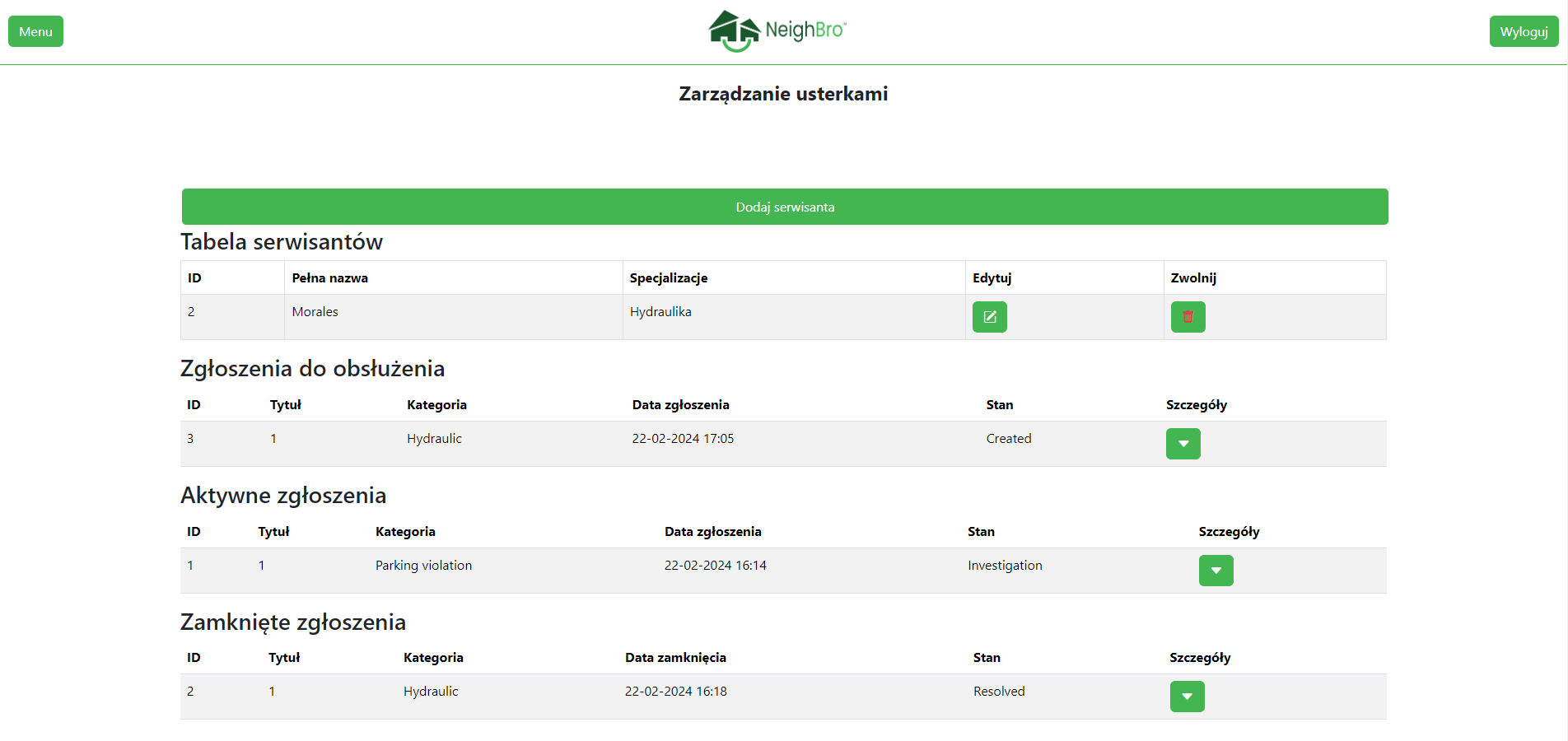


Walidacja na pola puste i datę(data rozpoczęcia nie może być większa niż zakończenia).

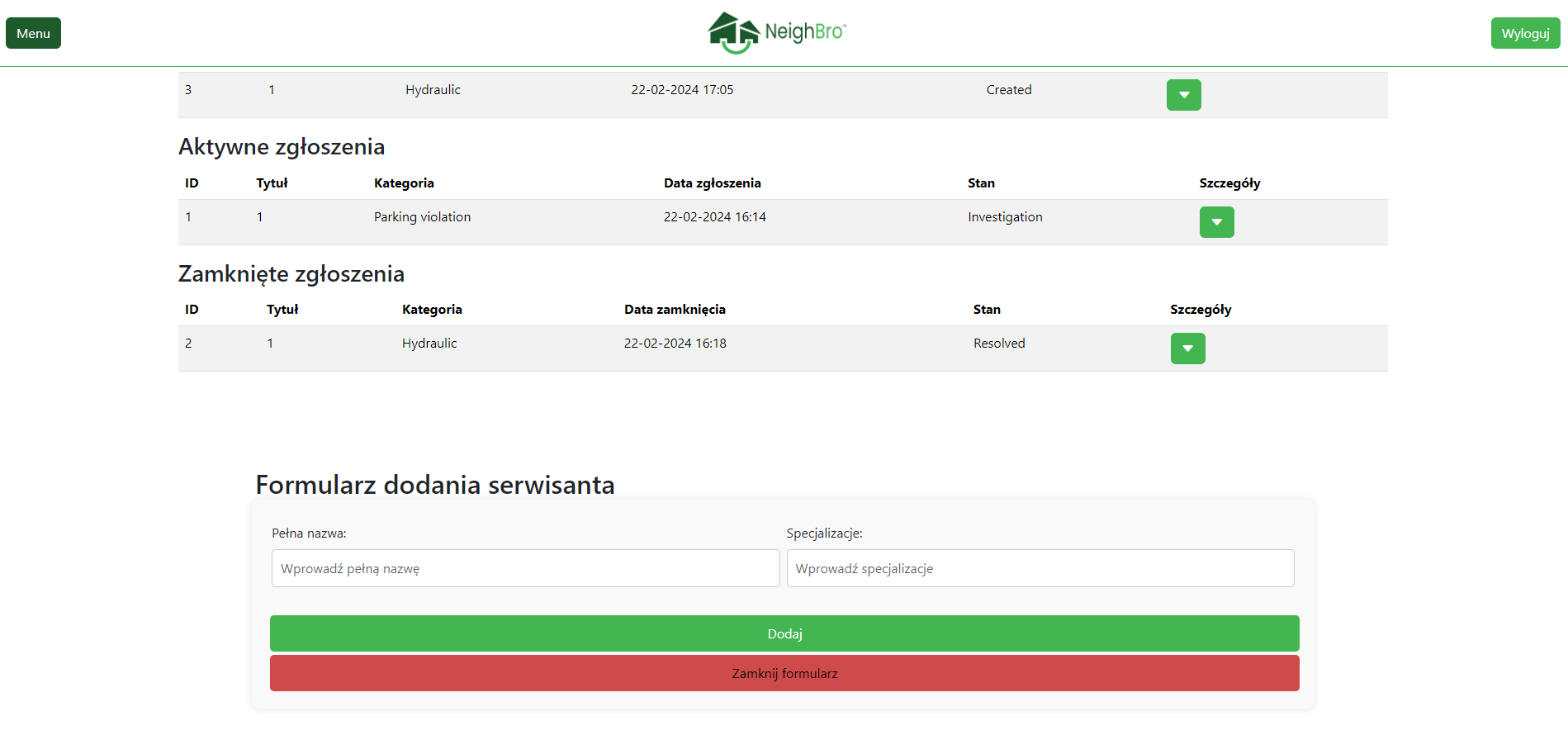
Administrator widzi głosowania podzielone na 3 sekcje nieaktywne(takie których data rozpoczęcia jest większa niż aktualna), aktywne(aktualna data jest w przedziale między początkiem a końcem glosowania) i zakończone.



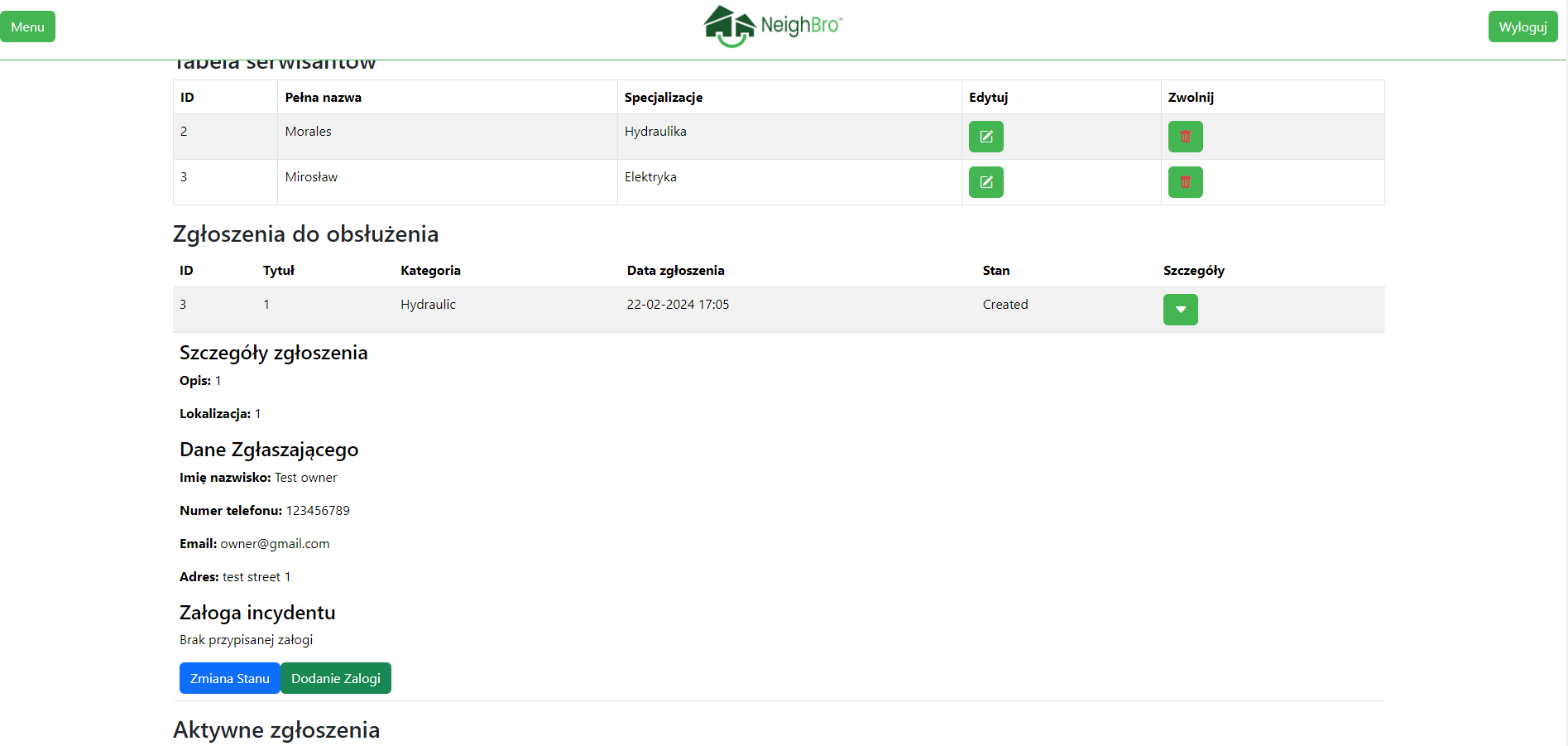
### Obsługa usterek

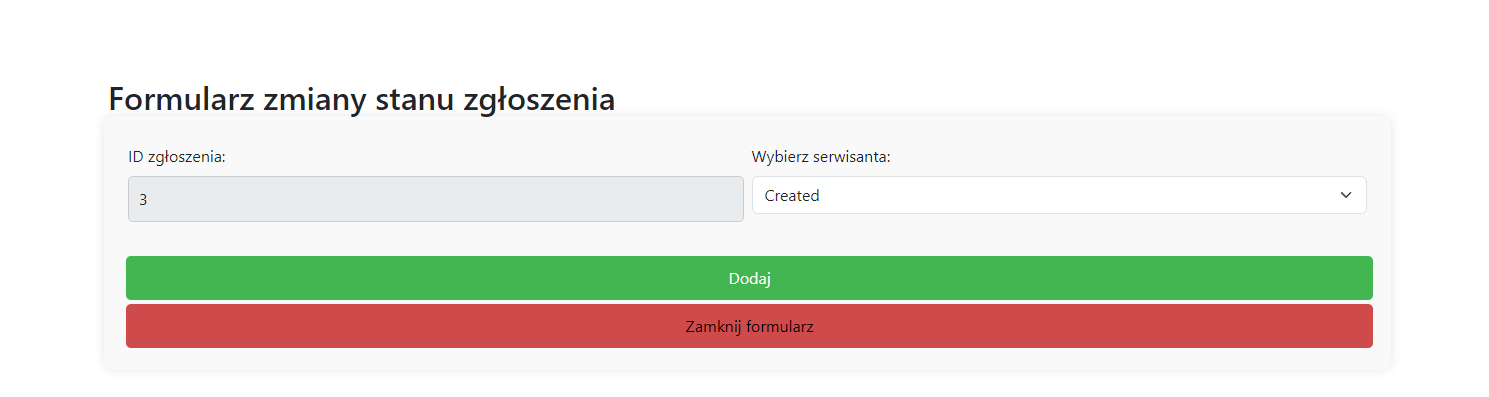


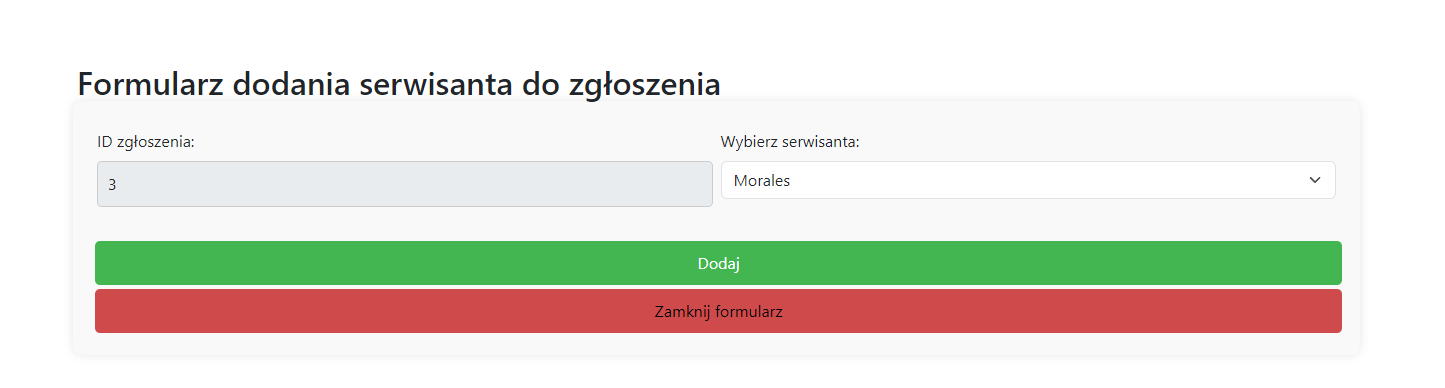
W tym panelu administrator może zarządzać usterkami oraz serwisantami. Przycisk „Dodaj serwisanta” umożliwia dodanie nowego serwisanta.



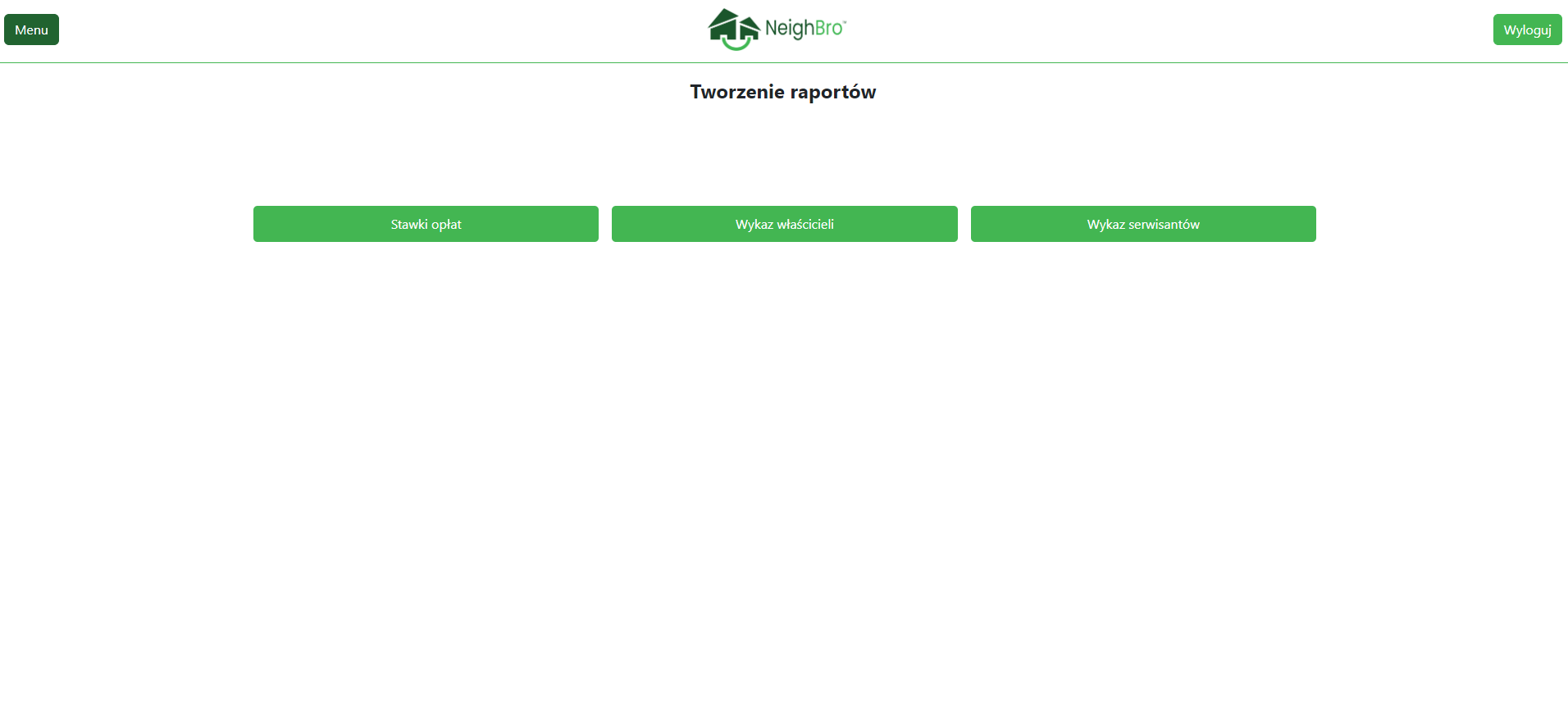
Zgłoszenia są grupowane na trzy kategorie w zależności od stanu do obsłużenia(nowe zgłoszenie stan created), aktywne(stan inny niż created i resolved) oraz zamknięte(stan resolved). Po kliknięciu przycisku „Szczegóły” zostaną wyświetlone szczegóły zgłoszenia takie jak opis i lokalizacja, dane zgłaszającego oraz załoga przypisana do incydentu. W tej pozycji możemy też zmienić stan zgłoszenia oraz dodać załogę.



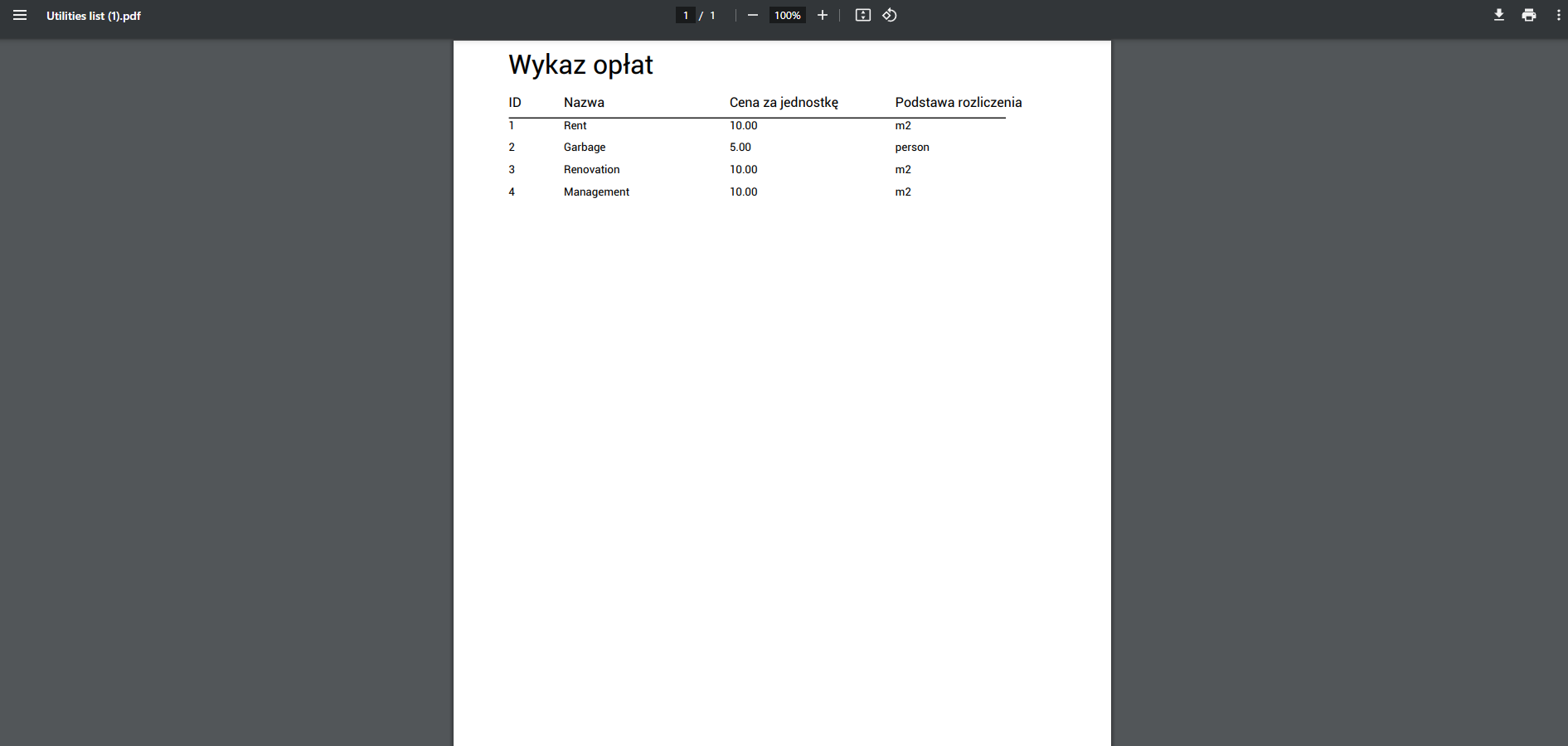




### Raporty



Sekcja ta umożliwia tworzenie raportów pdf z danych z bazy danych. Po kliknięciu w przycisk zostanie pobrany odpowiedni raport w formie PDF.

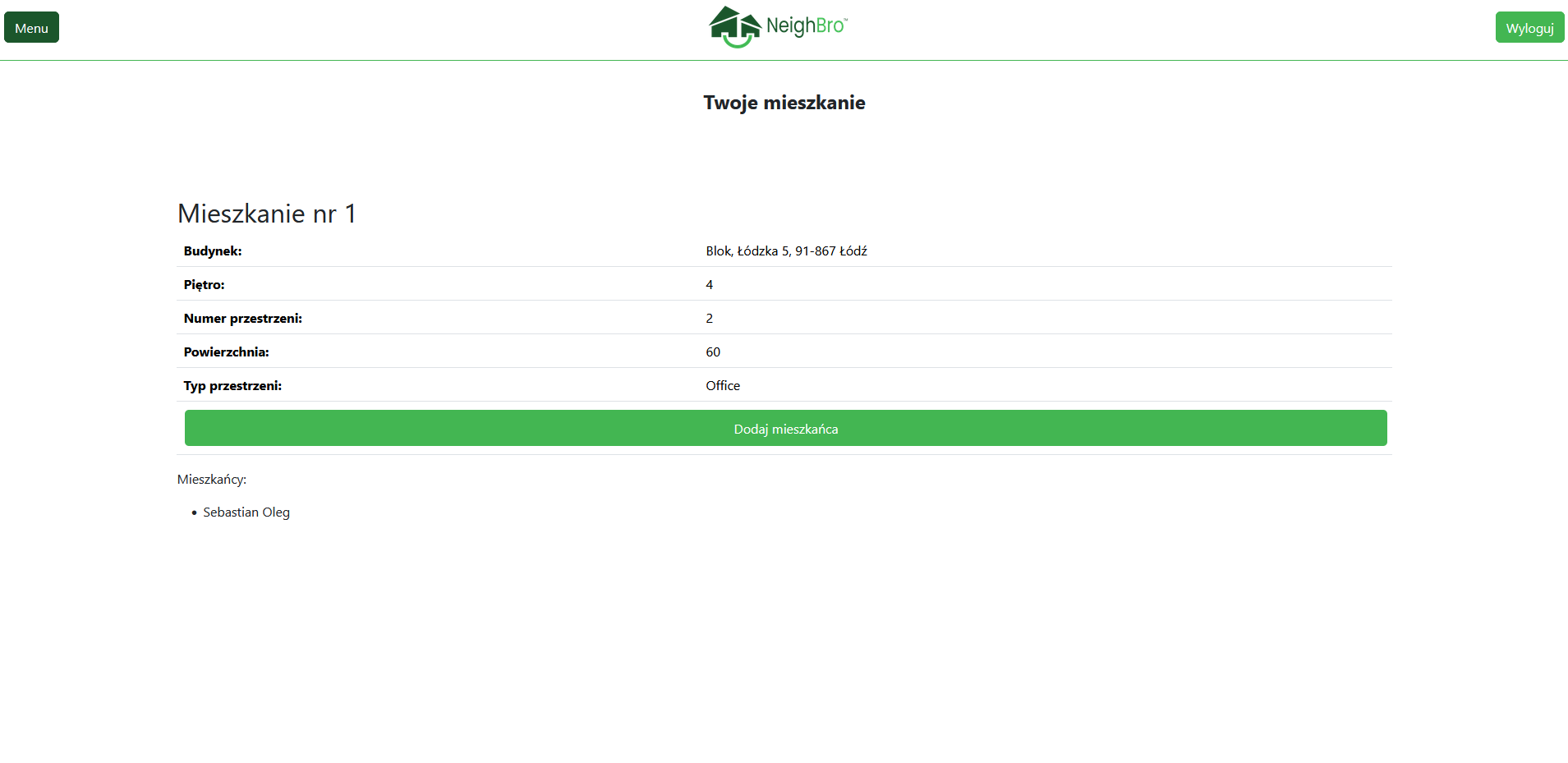


### Dokumentacja API

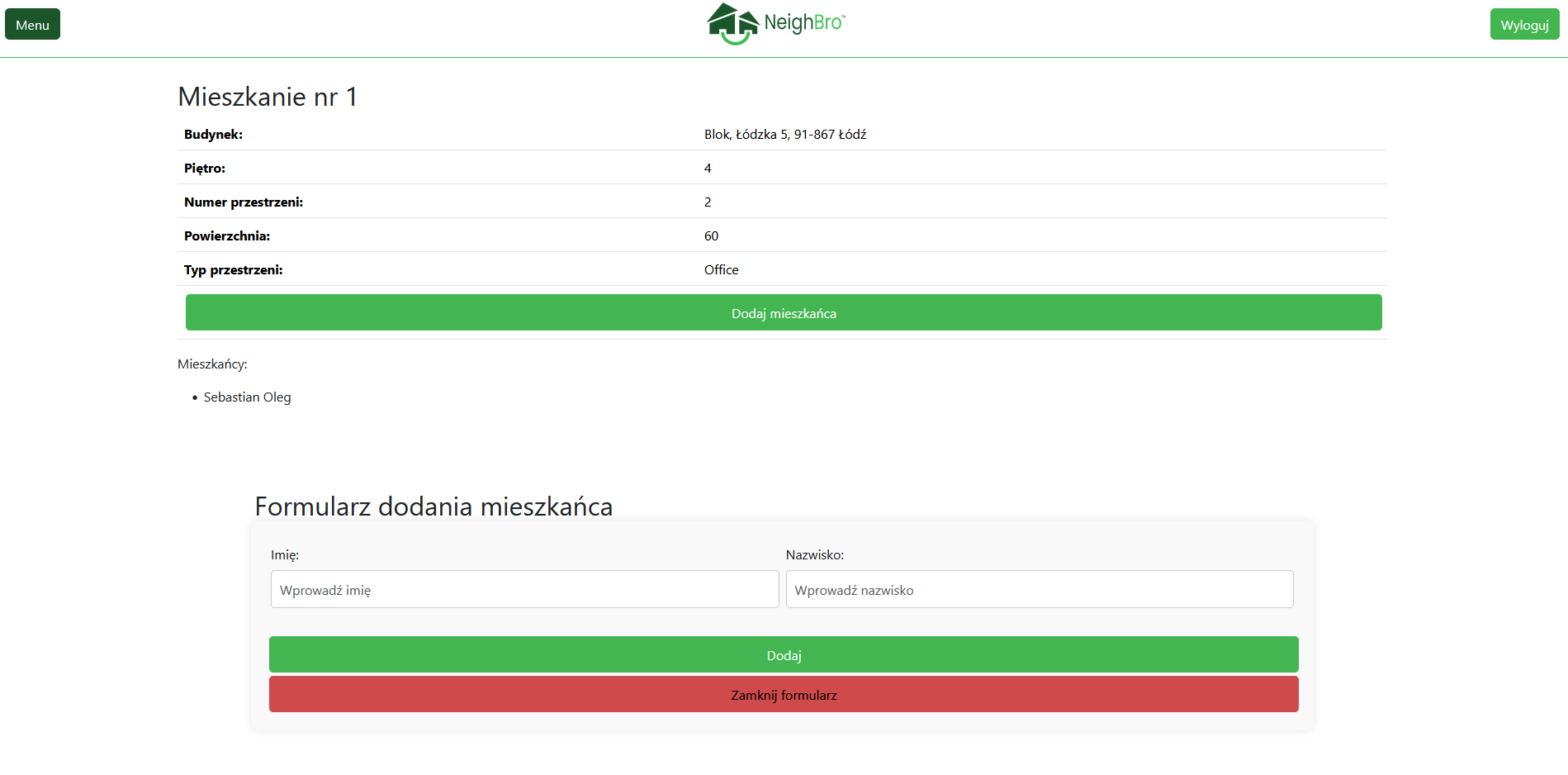
Opcja ta pozwala na podgląd requestów wysyłanych do API.



## Panel właściciela



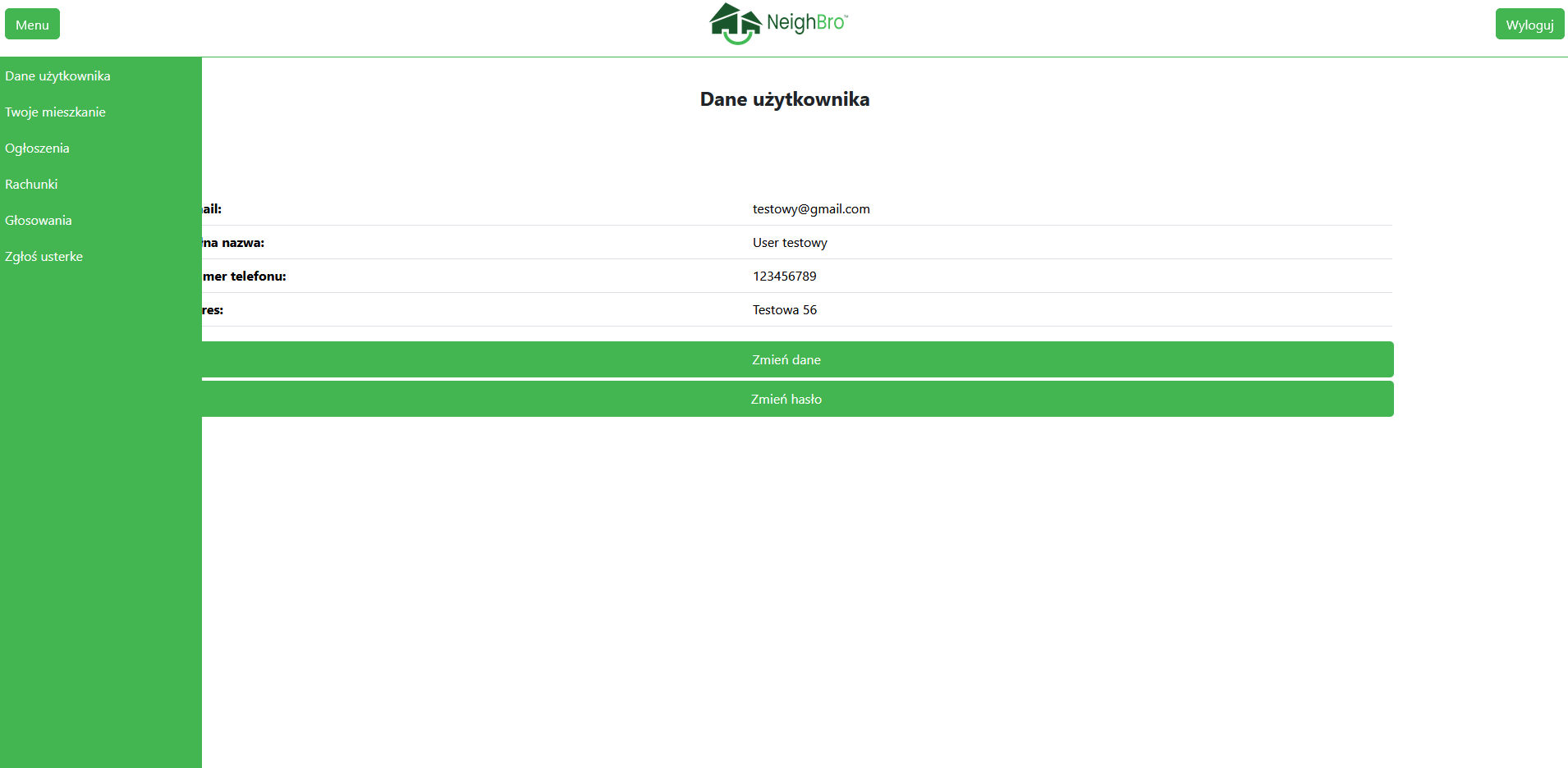
Po zalogowaniu się na konto właściciela zostanie mu pokazana lista jego mieszkań. Oraz opcja dodania mieszkańca do mieszkania.



W przypadku nie posiadania mieszkań zamiast listy zostanie wyświetlony komunikat.



### Menu



Menu właściciela posiada opcje takie jak:

-Dane użytkownika to samo co w przypadku panelu administratora

-Twoje mieszkanie strona startowa panelu użytkownika

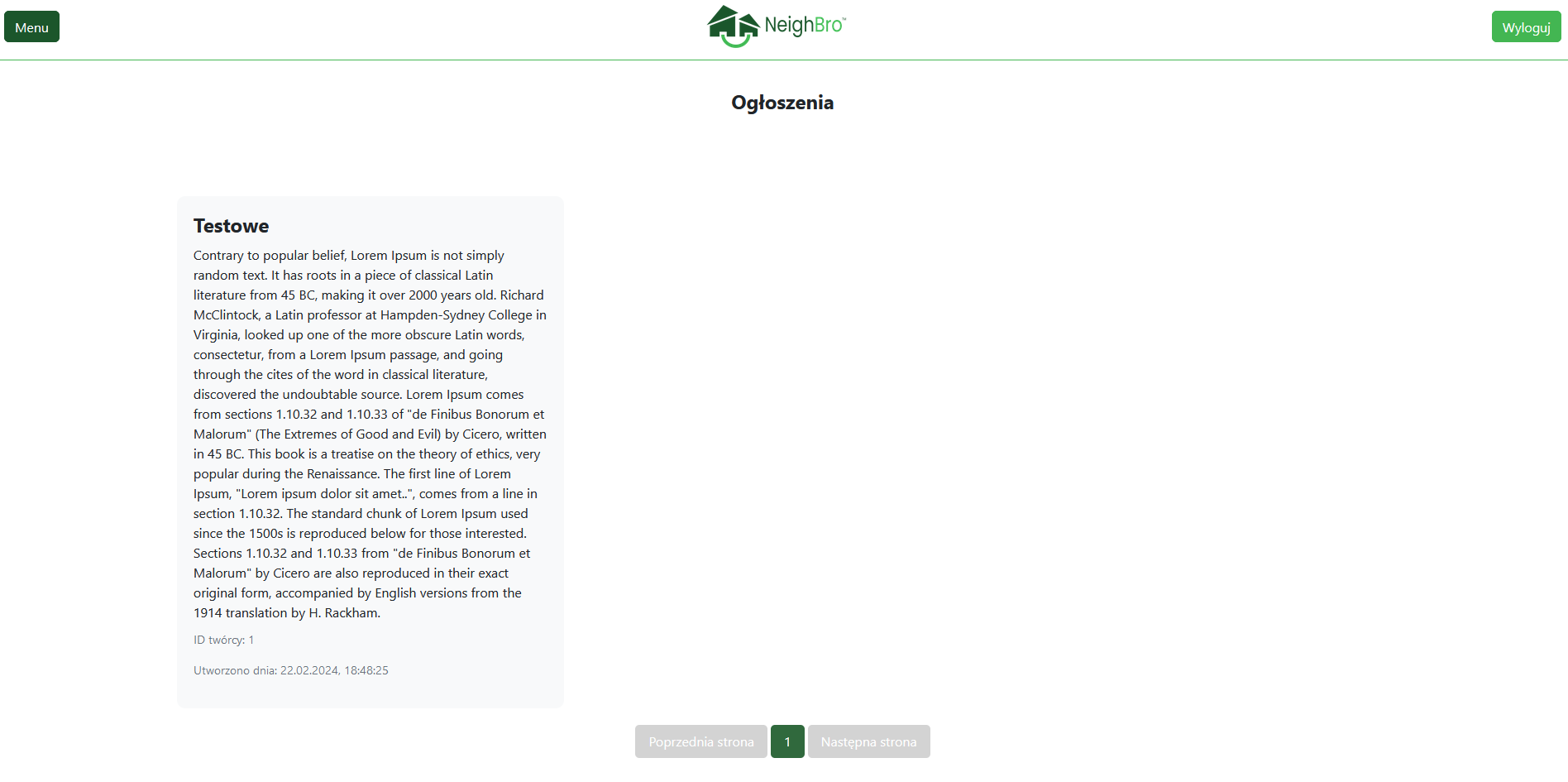
-Ogłoszenia podgląd aktualnych ogłoszeń

-Rachunki podgląd tabeli z opłatami i faktury na dany miesiąc

-Glosowanie udział w głosowaniach

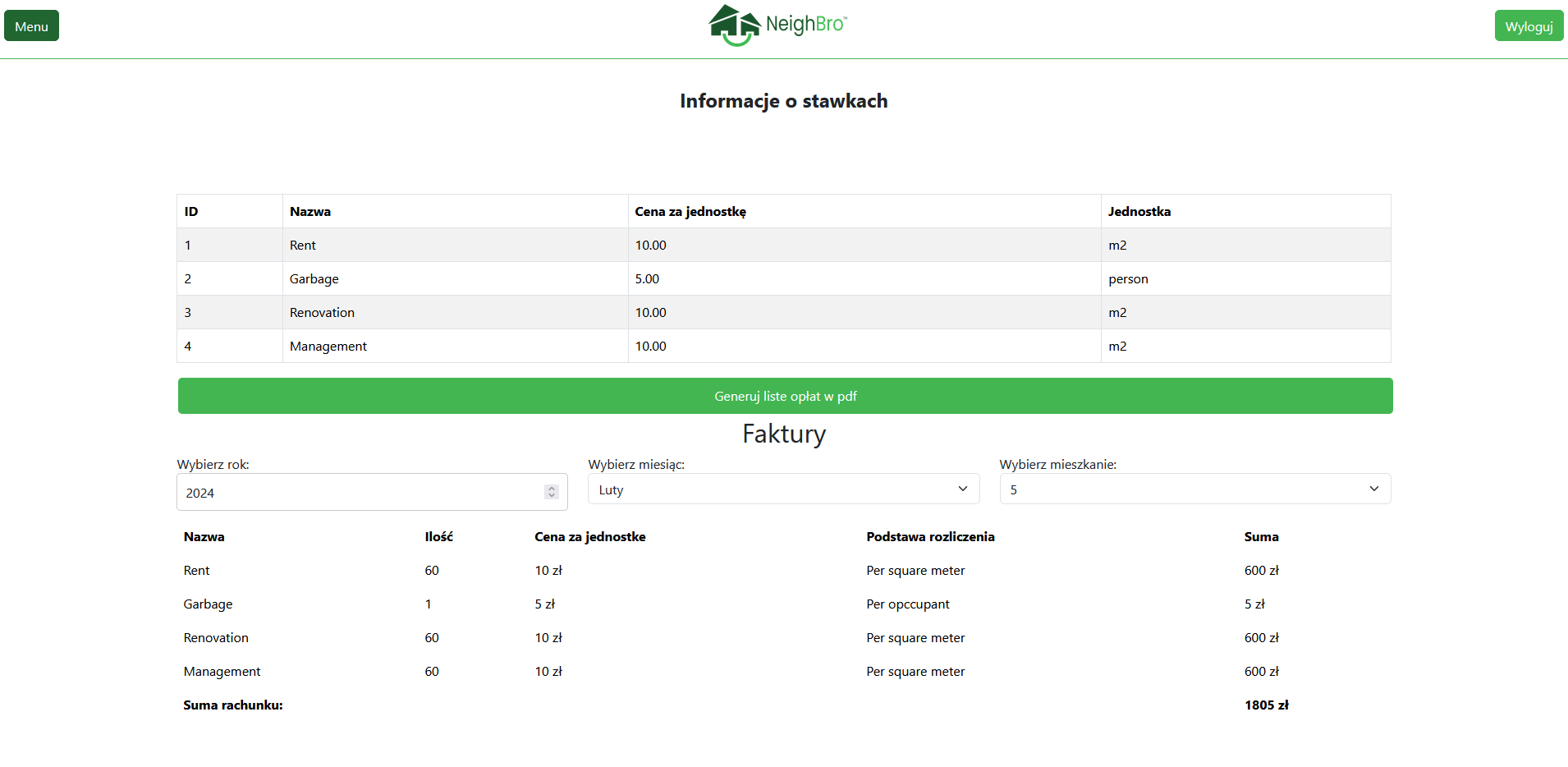
-Zgłoś usterek zgłaszanie i podgląd zgłoszonych usterek

### Ogłoszenia

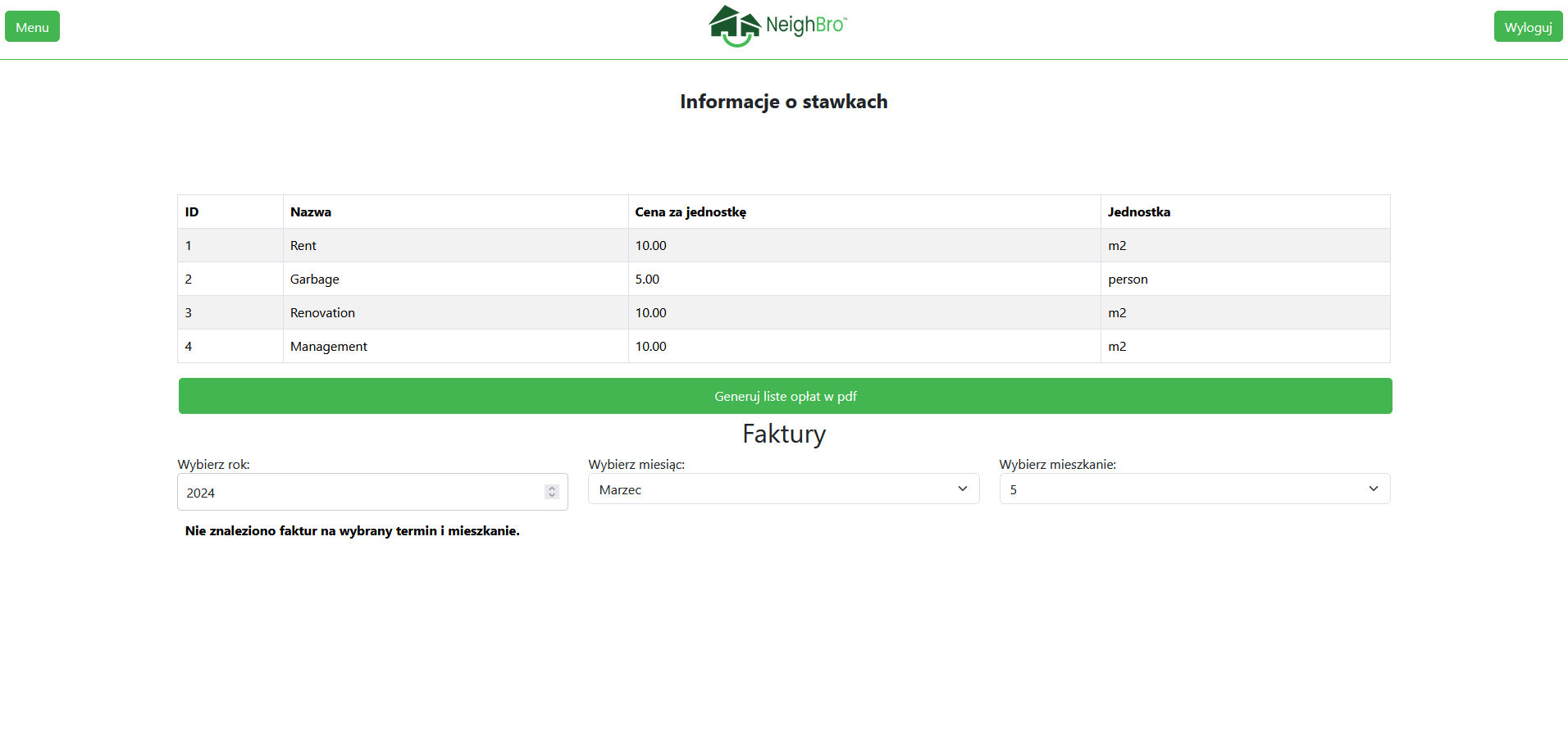


Opcja ta pozwala na podgląd ogłoszeń utworzonych przez administratora

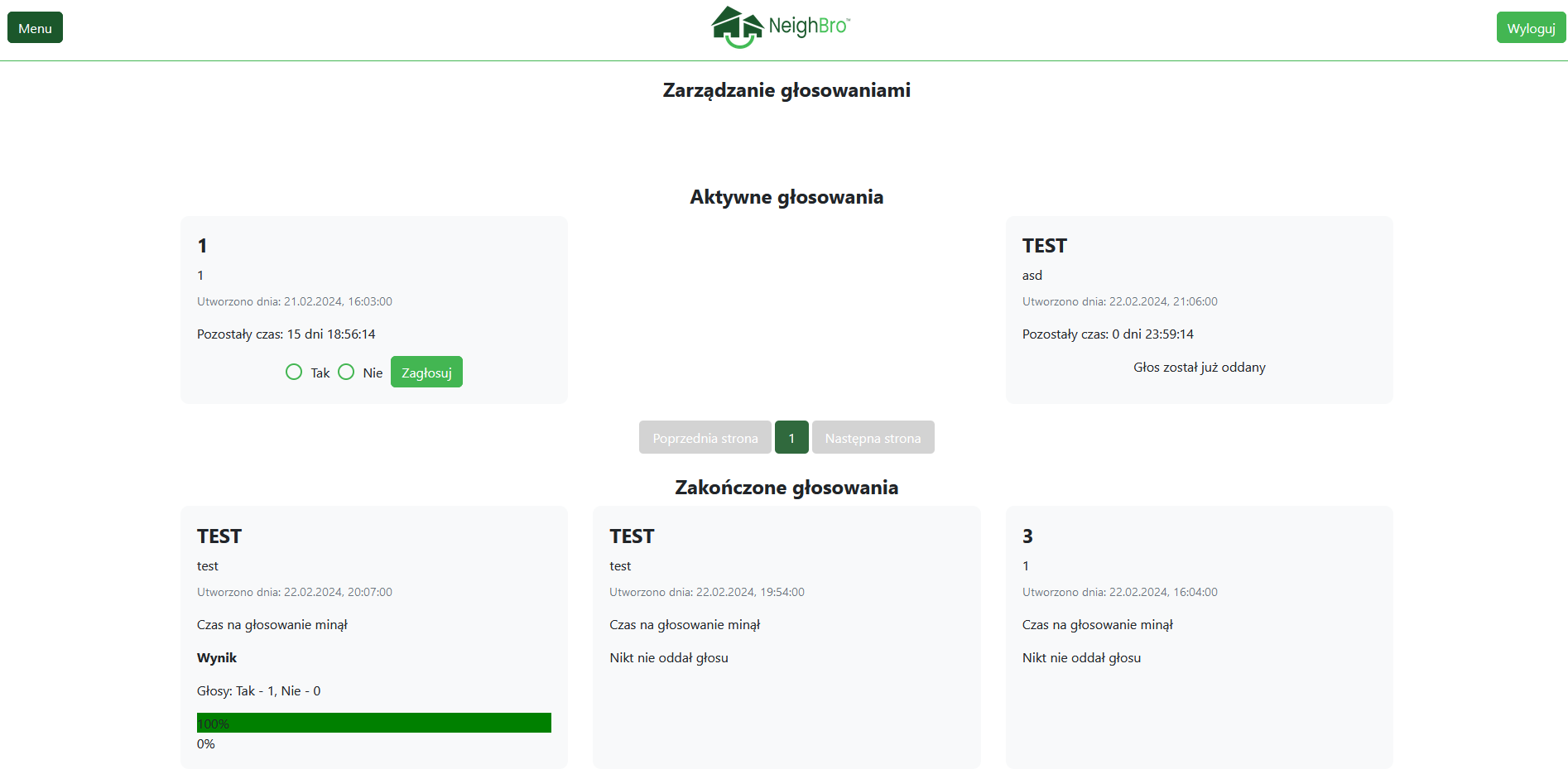
### Rachunki



Opcja ta umożliwia sprawdzenie aktualnej tabeli opłat oraz rachunków dla danego mieszkania za dany miesiąc oraz wygenerowanie tabeli z opłatami w formie PDF. Jeżeli właściciel posiada kilka mieszkań może on poprzez wybranie numeru mieszkania wyświetlić rachunek dla drugiego mieszkania. Jeżeli dla podanej daty niema faktury zostanie wyświetlony odpowiedni komunikat.

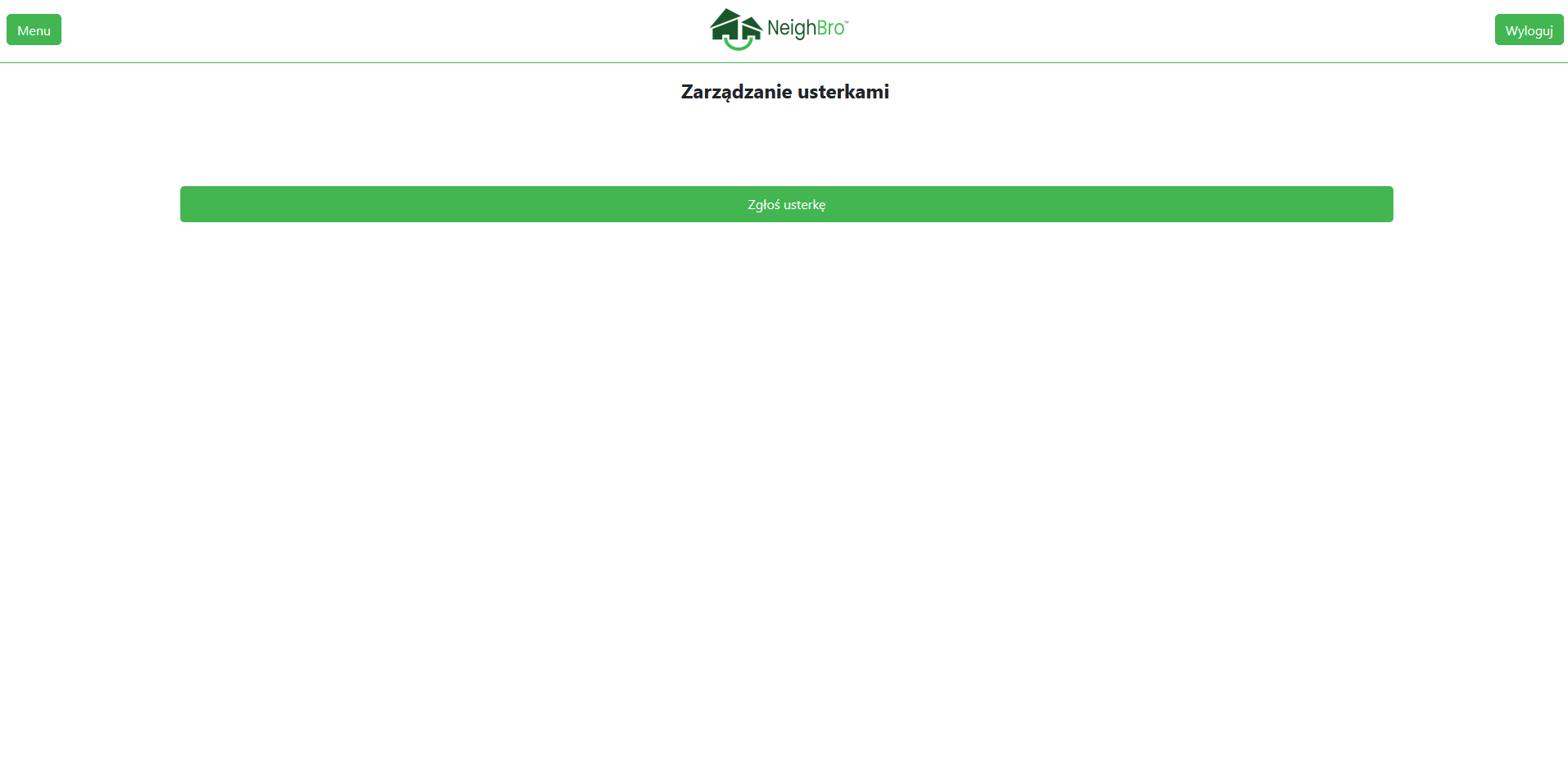


### Głosowania

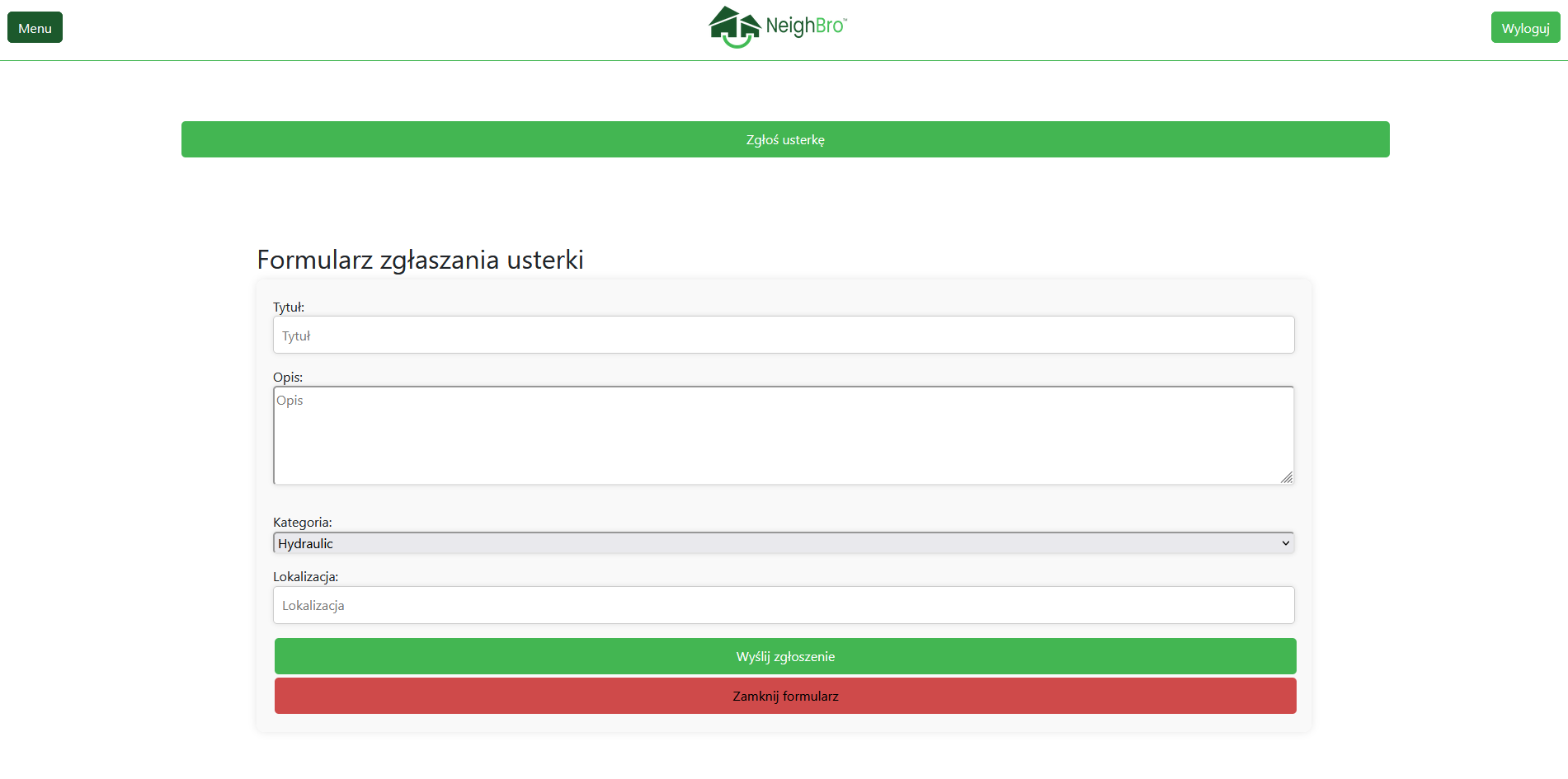


W tym panelu właściciel który posiad mieszkanie może oddać głos w glosowaniu oraz zobaczyć wyniki glosowań już zakończonych. Opcja głosowania blokuje się automatycznie o upłynięciu czasu na głosowania oraz gdy głos został już oddany.

### Zgłoś usterkę



W tej opcji właściciel może zgłosić zaistniała usterkę. Po kliknięciu przycisku „Zgłoś usterkę” zostanie wyświetlony formularz dodania usterki.



Lista kategorii jest generowana na podstawie kategorii w bazie danych. Walidacja na pola puste.

Po dodaniu usterki zostanie ona pokazana w tym panelu.



Po kliknięciu opcji szczegół zostaną wyświetlone szczegóły zgłoszenia takie jak opis i lokacja oraz załoga przypisana do incydentu.



Jeżeli zgłoszenie zostanie zamknięte zostanie ono przeniesione do listy zgłoszeń zamkniętych.

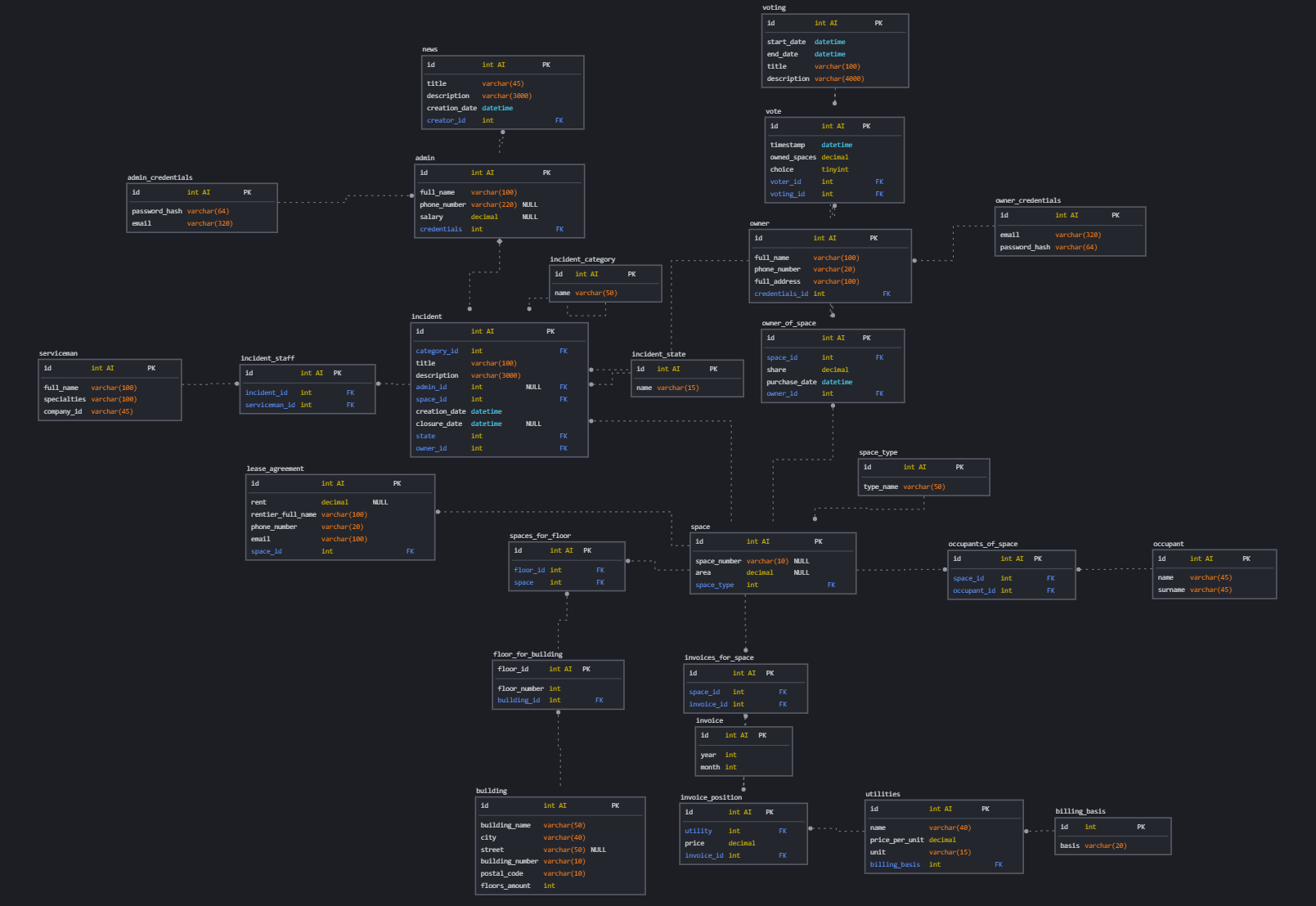


# Baza danych

## Opis

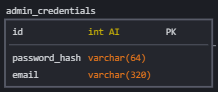
Do stworzenia bazy danych zostały wykorzystane MySQL oraz Docker. MySQL jest bardzo powszechną i dojrzałą technologią, co w połączeniu z Dockerem, daje przenośne i powtarzalne środowisko pracy.

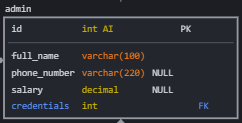
## Schemat bazy danych:

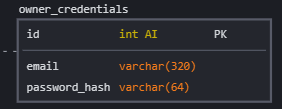


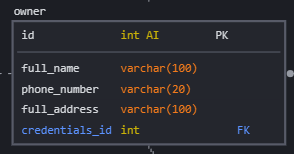
## Opis tabel w bazie danych

### Dane użytkowników:

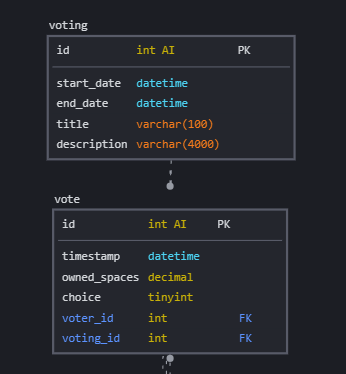




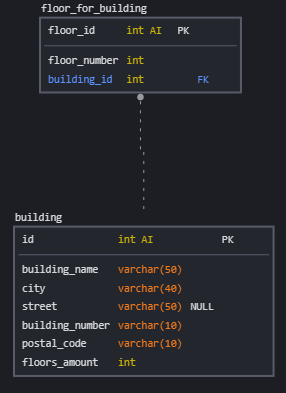




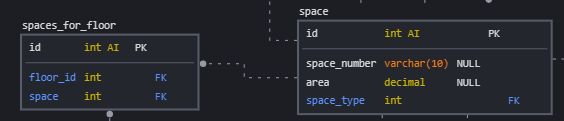
### Dane o głosowaniach:



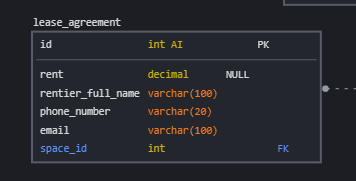
### Dane o budynkach:



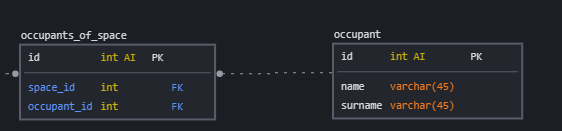
### Dane o przestrzeniach:



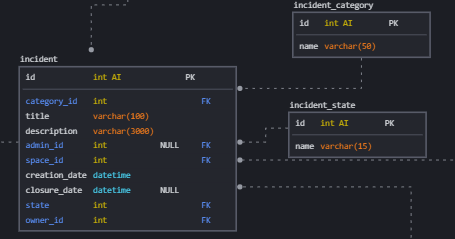
### Dane o umowach najmu:



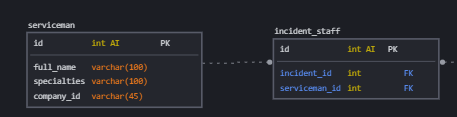
### Dane o mieszkańcach:



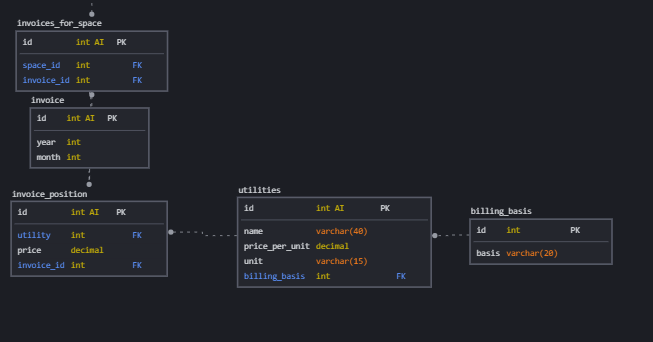
### Dane o incydentach:



### Dane o serwisantach:



### Dane o fakturach i opłatach:



# Backend

## Opis

Backend został stworzony w oparciu o technologię FastAPI, czyli framework Pythona 3. Jest to nabierająca rozpędu technologia backendowa, która zapewnia takie funkcjonalności, jak walidacja typów, obsługa modeli pydantic czy asynchroniczność wywołań.

Poszczególne endpointy zostały rozdzielone między “routery”, każdy zawierający tylko endpointy, odpowiadające za konkretną funkcjonalność, na przykład głosowania, incydenty, kreator osiedla, itp.

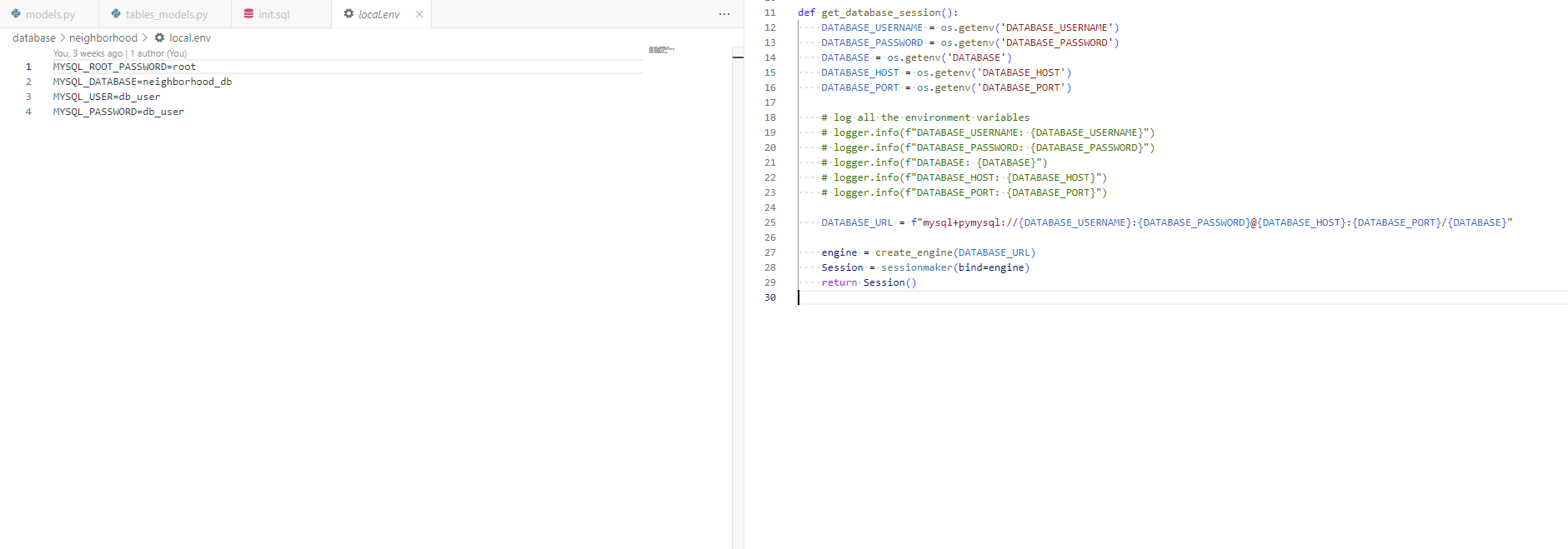
Fragment pliku spaces\_router.py:



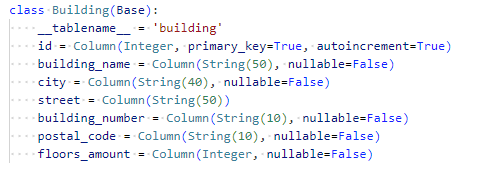
## Obsługa bazy danych

Operacje na bazie danych są wykonywane za pomocą sqlalchemy, narzędzie ORM (object-relational mapper), który pozwala na operowanie tabelami i rekordami w sposób zbliżony do operowania na obiektach.

Serwer operuje na bazie danych za pomocą konta stworzonego w pliku zmiennych środowiskowych bazy danych. Proces łączenia się z bazą danych:



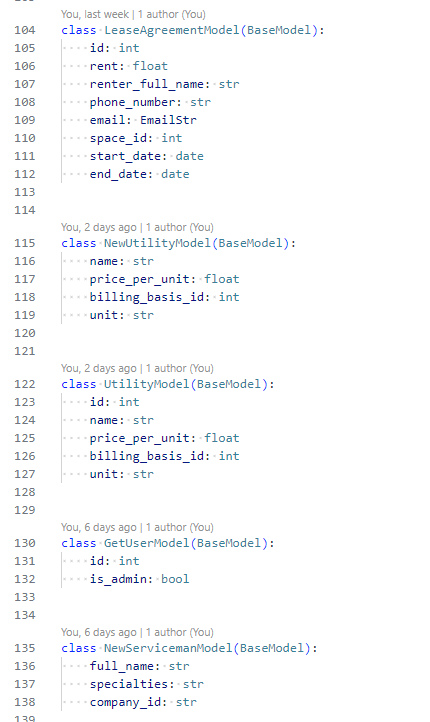
W celu korzystania z tego narzędzia, należało stworzyć modele każdej z tabel, uwzględniające pola, typy oraz relacje między tabelami. Przykładowy model ORM wykorzystany w projekcie:



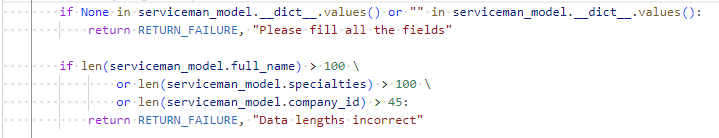
## Wykorzystanie programowania defensywnego

Programowanie defensywne to sposób tworzenia oprogramowania, który zakłada, że błędy, awarie i nieprzewidziane sytuacje będą miały miejsce, i w związku z tym aktywnie podejmuje kroki w celu ich obsługi.

Serwer stosuje wnikliwe sprawdzenie danych dla każdego zapytania. Sprawdzanie zaczyna się na modelach pydantic oraz wbudowanych funkcjonalnościach FastAPI. Zapytanie, które nie spełnia zadanej składni, zostanie automatycznie odrzucone. Przykładowe modele pydantic:



Następnie dane są sprawdzane pod kątem kompletności i poprawności. Przykład takiego zabezpieczenia w funkcji new\_serviceman():



Przed wykonaniem działań na bazie danych, sprawdzana jest również prawdziwość przesłanych danych. Zapobiega to operacjach na typie **None**, które mogą wystąpić przy nieostrożnej implementacji.

### Docker

Zostały stworzone 3 kontenery:

* Baza danych MySQL
* Serwer, aplikacja FastAPI
* Phpmyadmin - narzędzie wspomagające development

Uruchomienie tych trzech paczek jest koordynowane przez plik docker-compose.yml, i można je wyzwolić za pomocą skryptu start\_backend.bat. Skrypt aktywuje plik docker compose, lecz nie Docker Daemon-a, ten musi być włączony w chwili uruchamiania skryptu.

Konsola tego skrytpu jest również wspólną konsolą wszystkich zawartych usług, więc może służyć do podglądu zapytań oraz logów.