

**ETAP 4**

# **SYSTEM USŁUG SPRZĄTAJĄCYCH**

Imię i nazwisko studenta: **Jakub Owczarzak, Andrzej Chmielewski**  
**Rafał Spizewski, Oliwia Andres, Michał Mackiewicz**

Grupa: **Z710**

Specjalność: **Inżynieria Internetu**

## Spis treści

<b>1. Implementacja Bazy danych</b>	2
1.1 Konfiguracja bazy danych	3
1.2 Zarejestrowane zlecenia	3
1.3 Użytkownicy	4
<b>2. Implementacja warstwy logicznej</b>	5
2.1 Logika biznesowa	5
2.2 Architektura warstwowa	8
<b>3. Implementacja GUI</b>	9
3.1 Ekran główny dla osoby nie zalogowanej	9
3.2 Ekran logowania	9
3.3 Ekran rejestracji	10
3.4 Ekran zalogowanego użytkownika	10
3.5 Dodawanie miejsca wykonania usługi	11
3.6 Widok dodanego przez użytkownika miejsca wykonania usługi	11
3.7 Podgląd wybranego adresu na mapie	12
3.8 Edycja miejsca wykonania usługi	12
3.9 Usuwanie miejsca wykonania usługi	13
3.10 Widok strony głównej dla konta Admin	13
3.11 Widok zarejestrowanych użytkowników dla konta Admin	14
3.12 Podgląd zgłoszonych miejsc dla wybranego użytkownika z poziomu konta Admin	14
<b>4. Testy aplikacji</b>	15
4.1 Poprawna rejestracja do systemu	15
4.2 Poprawne logowanie do systemu	18
4.3 Sprawdzanie dodanych miejsc	20
4.4 Dodawanie nowych miejsc	20
4.4 Edycja dodanego miejsca	23
4.5 Usuwanie dodanego miejsca	24
4.6 Wylogowanie	25
4.7 Sprawdzenie listy użytkowników	25
4.8 Sprawdzenie profilu konkretnego użytkownika	26

## 1. Implementacja Bazy danych

## 1.1 Konfiguracja bazy danych

Konfiguracja bazy danych odbywa się w pliku server.js gdzie zachodzą wszystkie operacje związane z inicjalizacją podłączenia bazy danych do projektu wraz z obsługą back-endu przy użyciu biblioteki express.

```
const express = require('express');
const mongoose = require('mongoose');
const config = require('./server/config/config');
const HttpError = require('./model/error');
const path = require('path');

const fs = require('fs'); // biblioteka file system

const placeRouter = require('./server/routes/place-routes');
const authRouter = require('./server/routes/auth-routes');
const bodyParser = require('body-parser');

const app = express();

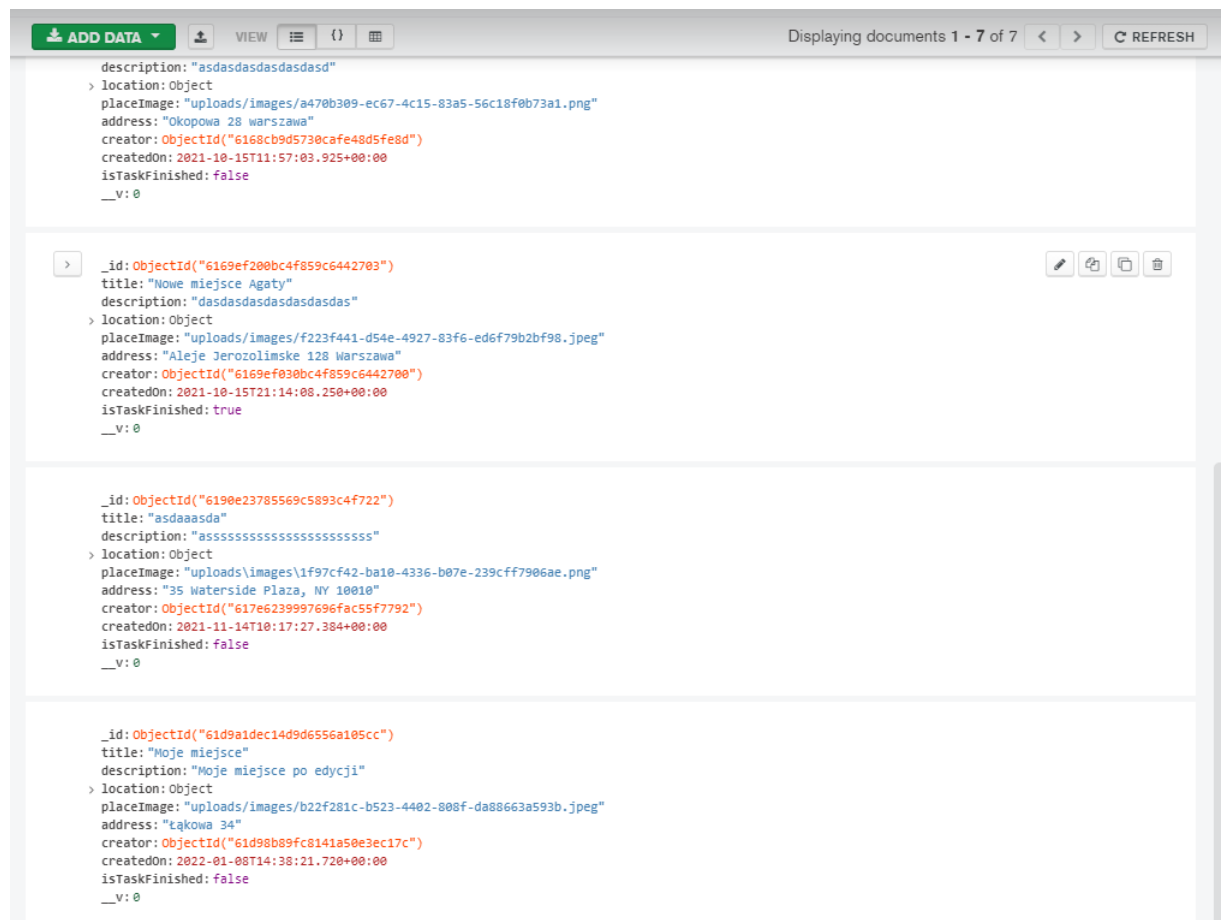
const port = process.env.PORT || 8080;

mongoose.connect(config.hostDB);
mongoose.connection.on('connected', () => {
  console.log('PLACES_APP database connected');
});
```

## 1.2 Zarejestrowane zlecenia

W bazie danych znajdują się informacje o zarejestrowanych miejscach wykonania usługi wraz z informacjami niezbędnymi do właściwej obsługi danych przez aplikację. W przykładowej pozycji dla miejsca wykonywanej usługi znajdują się takie informacje jak:

- tytuł zlecenia
- opis
- lokalizacja
- zdjęcie poglądowe (nie wymagane)
- autor zgłoszenia
- czas zarejestrowania
- flaga odpowiadająca za informacje czy zlecenie zostało zakończone

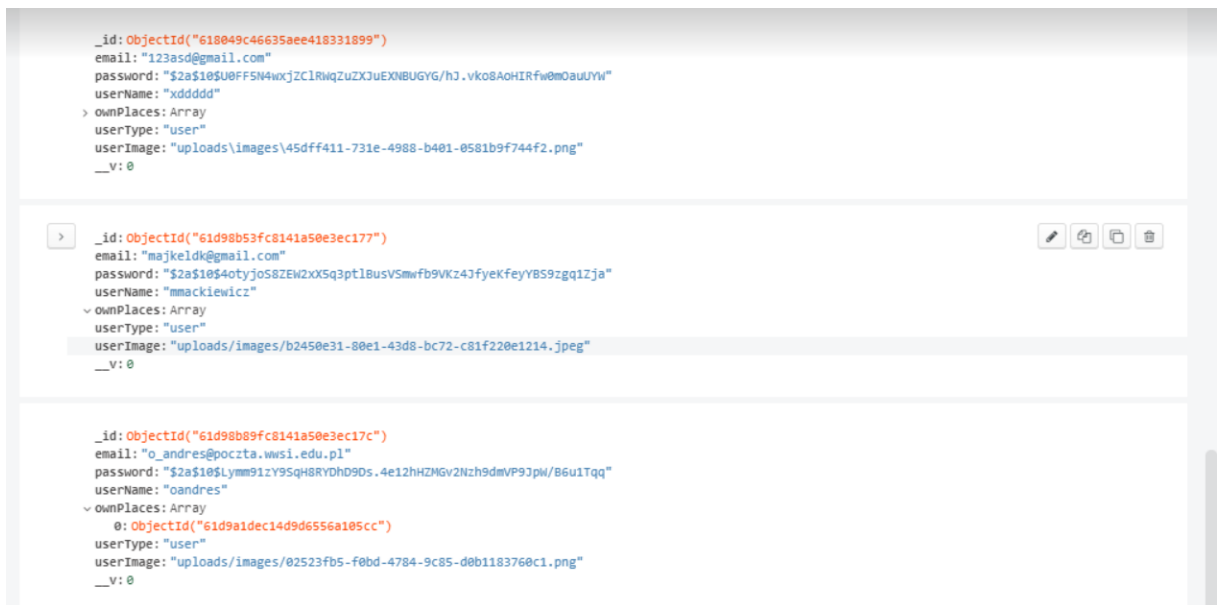


### 1.3 Użytkownicy

Baza przechowuje również informacje o zarejestrowanych w aplikacji użytkownikach. Aplikacja nie pozwala na zarejestrowanie zlecenia dla użytkowników nie posiadających konta. Informacje o

użytkownikach przechowywane w bazie zostały zawężone do najpotrzebniejszych w celu uzyskania jak najpłynniejszej komunikacji użytkownika z serwisem. Informacje które baza zbiera to:

- id użytkownika
- adres e-mail
- za-hashowane hasło dla konta
- nazwa użytkownika
- tablica miejsc które użytkownik zgłosił
- typ konta,
- zdjęcia przypisane do konta

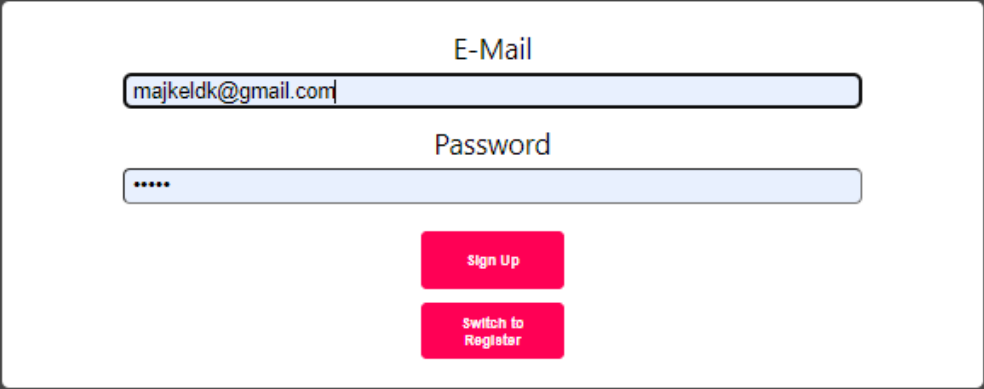


## 2. Implementacja warstwy logicznej

### 2.1 Logika biznesowa

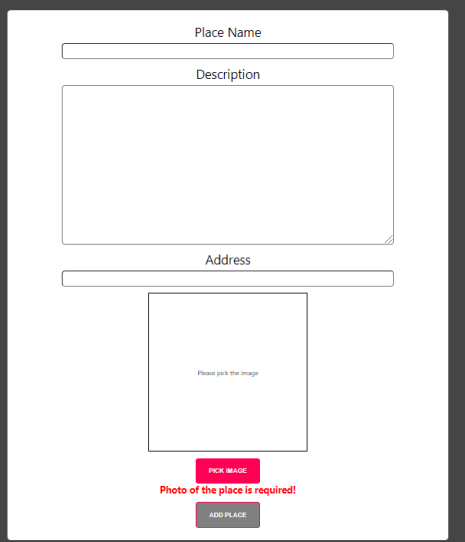
Logika biznesowa aplikacji polega na umożliwieniu użytkownikom zamówienia usług sprzątających w wybranym przez nich miejscu.

Użytkownik ma możliwość zalogowania do aplikacji:



The login form is centered on a dark gray background. It features two input fields: 'E-Mail' with the text 'majkeldk@gmail.com' and 'Password' with six dots. Below the fields are two red buttons: 'Sign Up' and 'Switch to Register'.

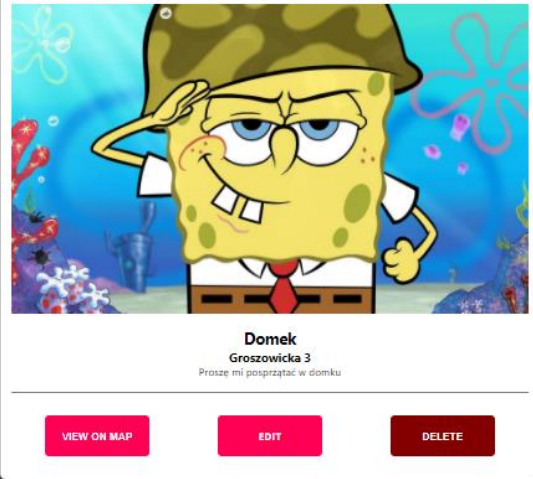
↓  
Dodania miejsca usługi



The form for adding a new service location is enclosed in a dark gray border. It contains four input fields: 'Place Name', 'Description', and 'Address'. Below these is a photo upload section with a placeholder image and the text 'Please pick the image'. At the bottom, there are two red buttons: 'PICK IMAGE' and 'ADD PLACE'. A red error message 'Photo of the place is required!' is displayed below the photo upload section.




Podglądu dodanego miejsca wraz z  
możliwością jego edycji, podglądu na  
mapie i usunięcia




The preview of a service location entry is shown in a white box with a dark gray border. It features a photo of SpongeBob SquarePants. Below the photo, the text reads 'Domek Groszowicka 3' and 'Proszę mi posprzątać w domku'. At the bottom, there are three red buttons: 'VIEW ON MAP', 'EDIT', and 'DELETE'.

Do dyspozycji oddajemy również konto admina z możliwością podglądu kont użytkowników, dodanych przez nich lokalizacji i zatwierdzania złożonych zleceń.



**Kubson**  
2 Places




**kubaliński**  
2 Places

**Paweł**  
0 Places




All Users



**WWSI SZKOŁA**  
Józefa Lewartowskiego 17 00-109 Warszawa  
Przeżył o porzucenie dla dla dla

VIEW ON MAP

DELETE



**Nowe miejsce**  
Jana Pawła II 27 Warszawa  
Miejscu Warszawa 0

VIEW ON MAP

EDIT

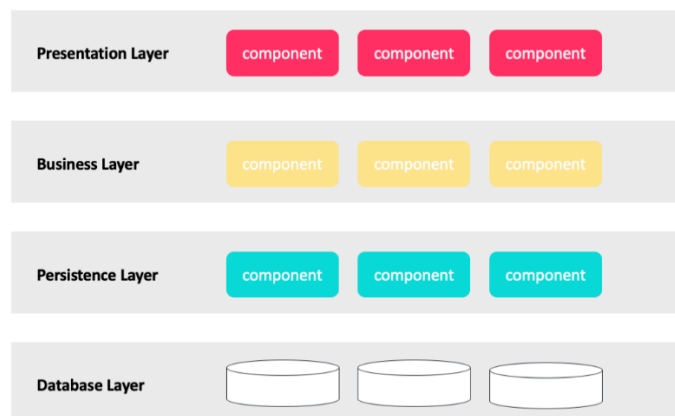
DELETE

## 2.2 Architektura warstwowa

Gdy złożoność aplikacji rośnie, jednym ze sposobów zarządzania tym złożonością jest podział aplikacji zgodnie z jej zadaniami lub problemami. Takie podejście jest oparte na zasadzie separacji problemów i może pomóc w uporządkowaniu rosnącej bazy kodu.

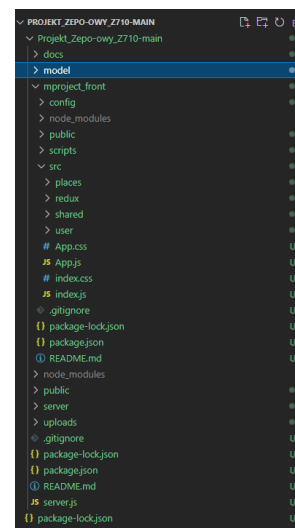
Dzięki zorganizowaniu kodu w warstwy można ponownie używać typowych funkcji niskiego poziomu w całej aplikacji. To ponowne użycie jest korzystne, ponieważ oznacza, że należy napisać mniej kodu, a ponieważ może to umożliwić aplikacji standaryzację w ramach jednej implementacji, zgodnie z zasadą nie powtarzaj się samodzielnie.

W przypadku aplikacji „Clean Company” zastosowano architekturę warstwową aplikacji dzięki czemu aplikacja ma możliwość wymuszenia ograniczeń, na których warstwy mogą komunikować się z innymi warstwami. Ta architektura pomaga osiągnąć hermetyzację. Zmiana lub zastąpienie warstwy powinno mieć wpływ tylko na te warstwy, które z nim pracują. Ograniczając, które warstwy zależą od innych warstw, można ograniczyć wpływ zmian, tak aby pojedyncza zmiana nie wpływała na całą aplikację.



Warstwy (i hermetyzacja) znacznie ułatwiają zastępowanie funkcji w aplikacji. Na przykład aplikacja może początkowo używać własnej bazy danych usługi SQL Server do utrwalania, ale później może użyć strategii trwałości opartej na chmurze lub jednej z internetowym interfejsem API.

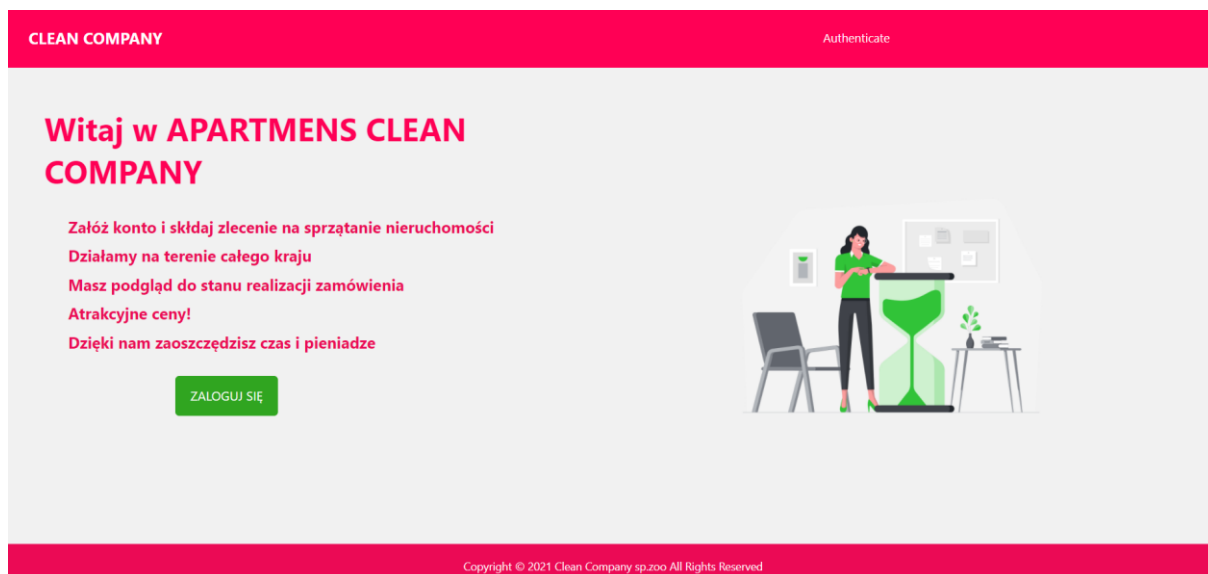
Oprócz możliwości zamiany implementacji w odpowiedzi na przyszłe zmiany wymagań, warstwy aplikacji mogą również ułatwić zamianę implementacji na potrzeby testowania.



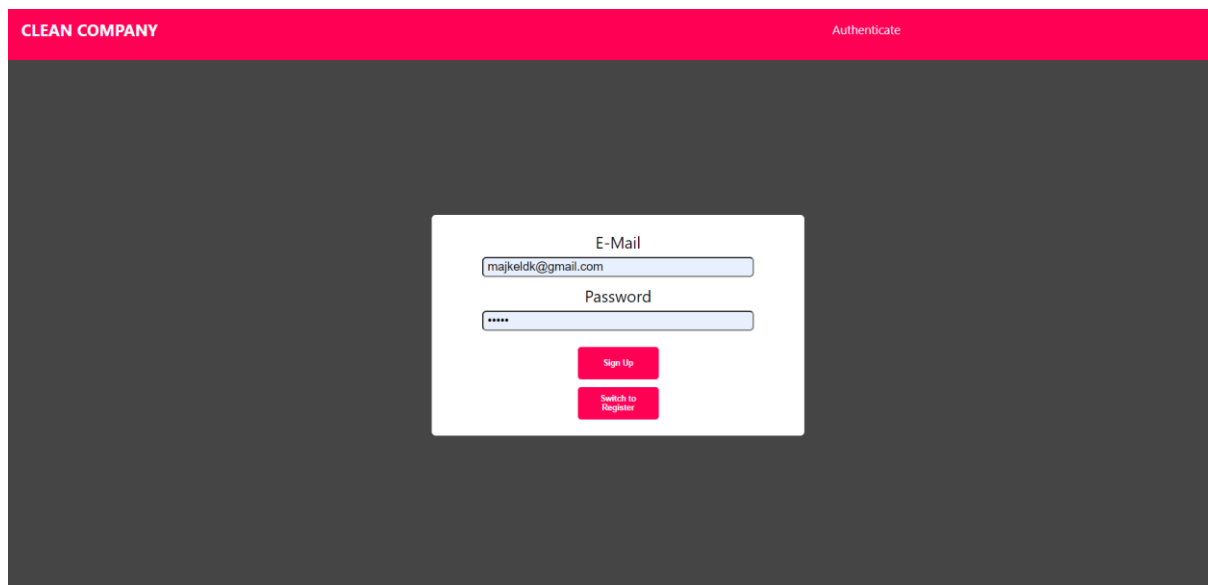


## 3. Implementacja GUI

### 3.1 Ekran główny dla osoby nie zalogowanej



### 3.2 Ekran logowania



### 3.3 Ekran rejestracji

CLEAN COMPANY

Authenticate

User Name

E-Mail

Required Password

Required Confirm Password

Profile Image

Please pick the image

PICK IMAGE

Register

Switch to Sign up

### 3.4 Ekran zalogowanego użytkownika

CLEAN COMPANY

Add Place

My Place

LogOut

**Witaj w APARTMENS CLEAN COMPANY**

Załącz konto i składaj zlecenie na sprzątanie nieruchomości


Działamy na terenie całego kraju

Masz podgląd do stanu realizacji zamówienia

Atrakcyjne ceny!

Dzięki nam zaoszczędzisz czas i pieniądze

ZALOGUJ SIĘ



Copyright © 2021 Clean Company sp.zoo All Rights Reserved

### 3.5 Dodawanie miejsca wykonania usługi

CLEAN COMPANY

Add PlaceMy PlaceLogOut

Place Name

Description

Address

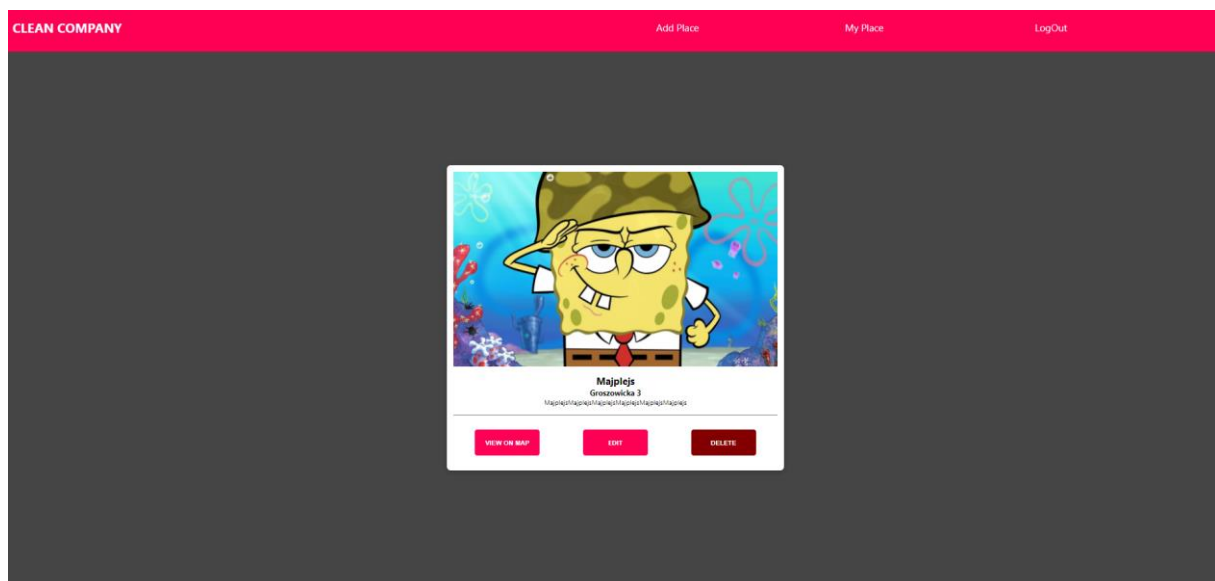
Please pick the image

PICK IMAGE

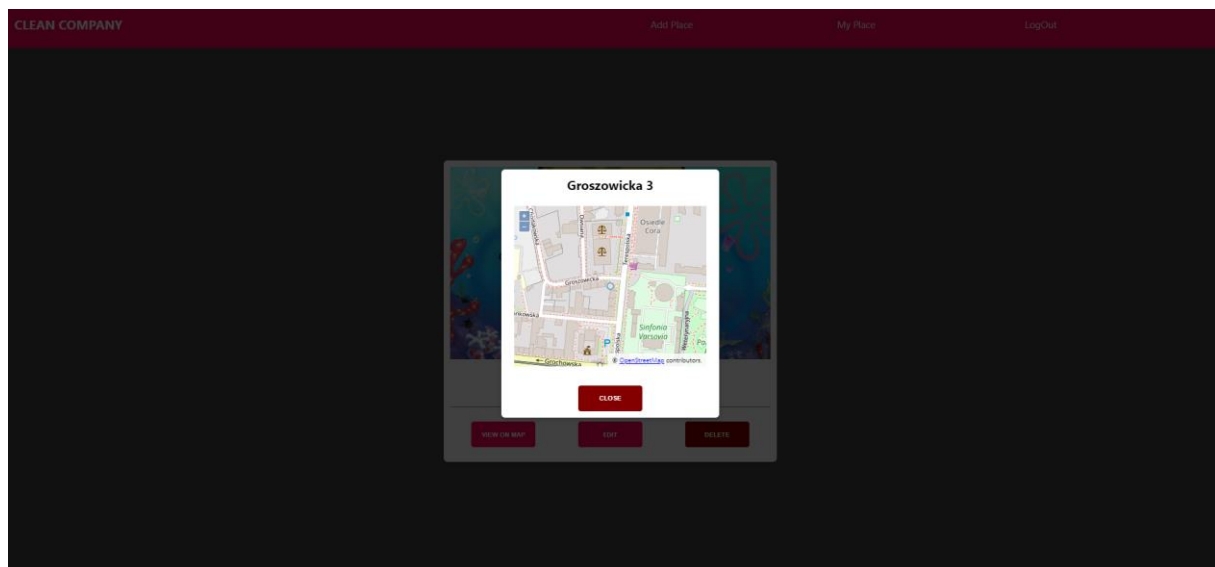
Photo of the place is required!

ADD PLACE

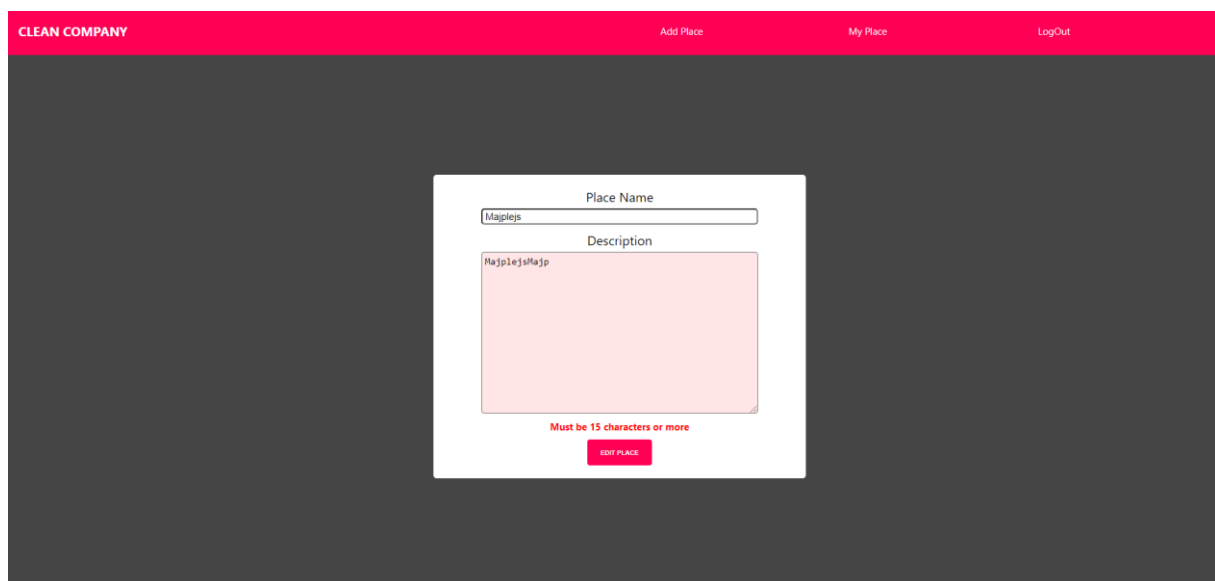
### 3.6 Widok dodanego przez użytkownika miejsca wykonania usługi



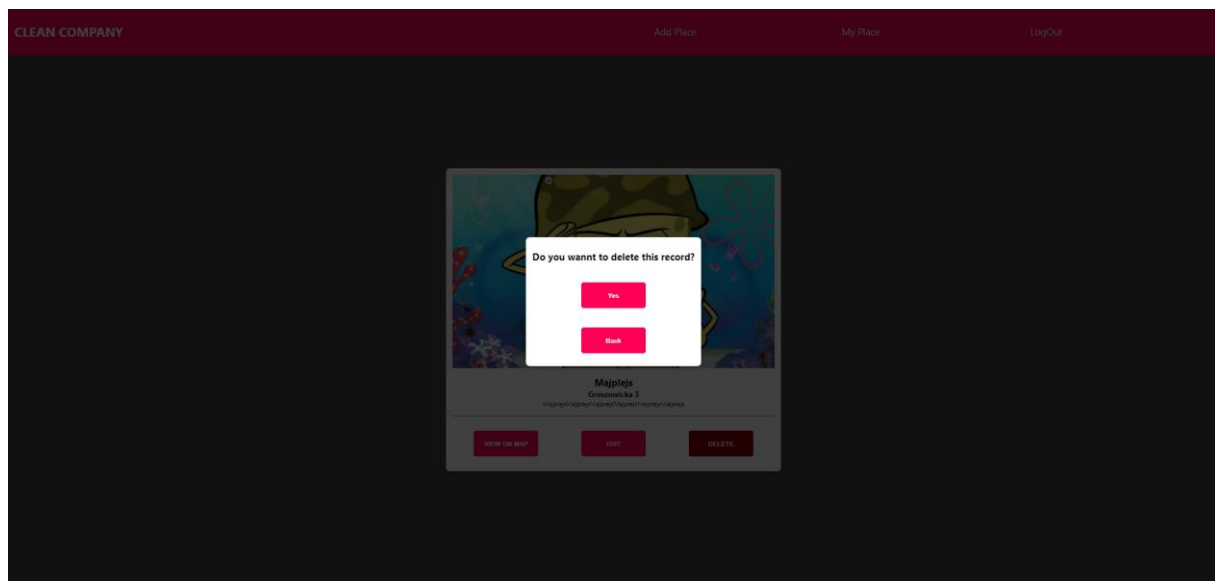
### 3.7 Podgląd wybranego adresu na mapie



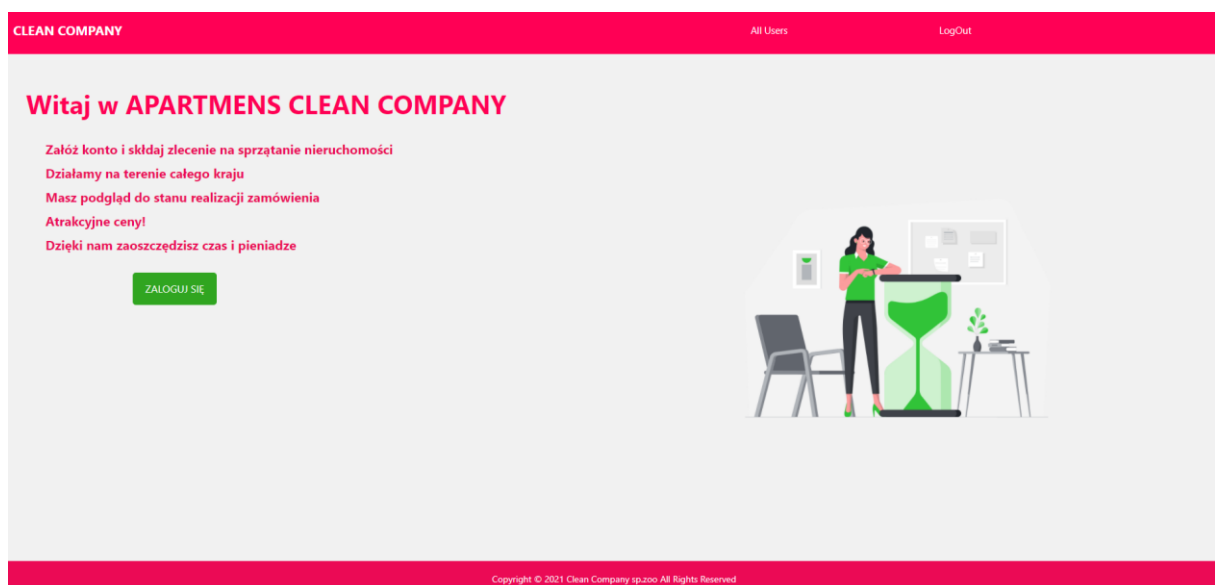
### 3.8 Edycja miejsca wykonania usługi



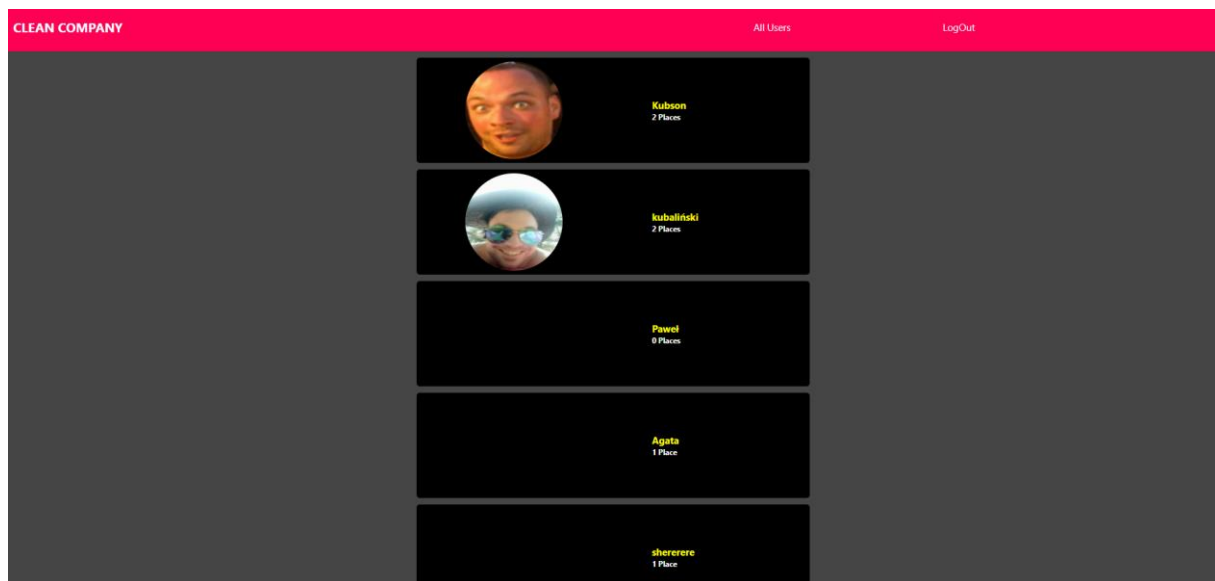
### 3.9 Usuwanie miejsca wykonania usługi



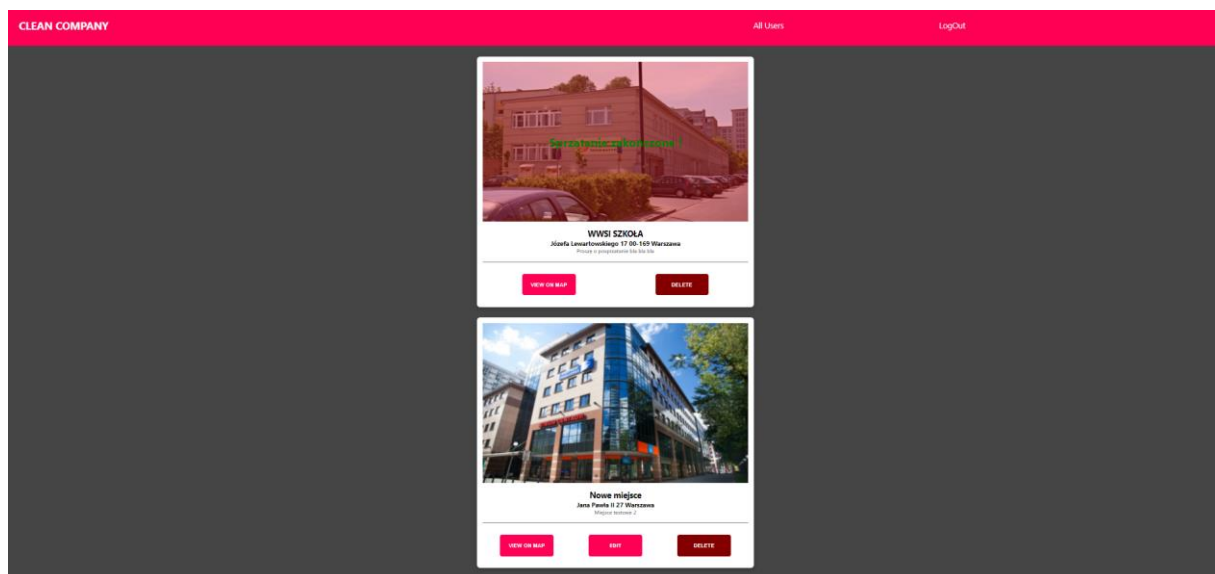
### 3.10 Widok strony głównej dla konta Admin



### 3.11 Widok zarejestrowanych użytkowników dla konta Admin



### 3.12 Podgląd zgłoszonych miejsc dla wybranego użytkownika z poziomu konta Admin



## 4. Testy aplikacji

### 4.1 Poprawna rejestracja do systemu

**Warunek wstępny:** Niezarejestrowany użytkownik, bez aktywnego konta w aplikacji.

**Kroki reprodukcji:**

1. Wybranie opcji „Zaloguj się”
2. Wybranie opcji „Przejdź do rejestracji”
3. Wprowadzenie loginu
4. Wprowadzenie adresu e-mail
5. Wprowadzenie hasła
6. Dodanie zdjęcia

**Oczekiwany rezultat:** Zarejestrowanie się do systemu jako nowy użytkownik

**Warunki końcowe:** Zarejestrowanie do systemu nowego użytkownika, przejście do ekranu głównego.

**Dane testowe:** *oandres*, *o\_andres@ms.wysi.edu.pl*, *12345*, *12345*

**Negatywne przypadki:**

#### 1. Błędny adres e-mail

**Oczekiwany rezultat:** Brak możliwości rejestracji do systemu, odpowiedni komunikat naprowadzający użytkownika.

**Warunki końcowe:** Przy błędnych danych w polu e-mail otrzymujemy informacje o konieczności poprawienia danych. Przy poprawnych brak komunikatu. Przy braku danych otrzymujemy informacje o konieczności wprowadzenia danych.

**Dane testowe:** Błędne: *o\_andresms.wysi.edu.pl*, Poprawne: *o\_andres@ms.wysi.edu.pl*

The image displays three sequential screenshots of a web registration form, illustrating different validation states:

- Left Screenshot:** Shows an error message "Incorrect format of email address" in red text below the E-Mail field. The Password and Confirm Password fields are marked as "Required" with red text.
- Middle Screenshot:** Shows the Password and Confirm Password fields marked as "Required" with red text. The E-Mail field is now empty.
- Right Screenshot:** Shows the Confirm Password field marked as "Required" with red text. The Password field is now empty.

Each screenshot includes the following fields and buttons:

- User Name:
- E-Mail: - Password: - Confirm Password: - Profile Image:
- Buttons: PICK IMAGE, Register, Switch to Sign up

## 2. Błędne hasło, błędne powtórzenie hasła

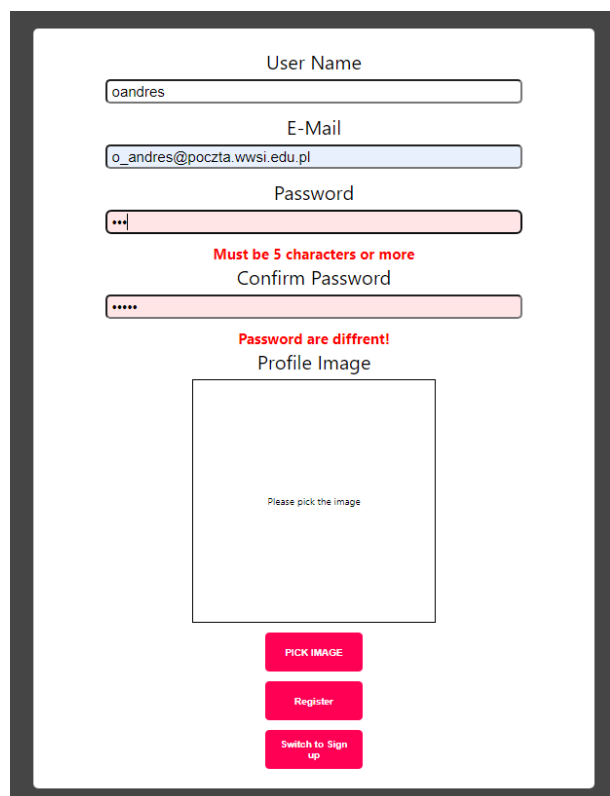
**Warunek wstępny:** Ustawienie hasła o minimalnej ilości znaków równej 5. Wpisanie dwóch takich samych haseł.

**Oczekiwany rezultat:** Brak możliwości rejestracji do systemu, odpowiedni komunikat naprowadzający użytkownika.

**Warunki końcowe:** Przy błędnych danych w polu hasło otrzymujemy informacje o konieczności poprawienia danych, przy błędnych danych w polu powtórz hasło otrzymujemy informacje o konieczności poprawienia danych. Przy poprawnych brak komunikatu. Przy braku danych otrzymujemy informacje o konieczności wprowadzenia danych.

**Dane testowe:** Błędne: 123, Poprawne: 12345

Odpowiednie komunikaty przy niepoprawnych danych.



The screenshot shows a registration form with the following fields and messages:

- User Name:** Input field containing "oandres".
- E-Mail:** Input field containing "o\_andres@poczta.wysi.edu.pl".
- Password:** Input field containing "123". Below it, a red error message reads: "Must be 5 characters or more".
- Confirm Password:** Input field containing "12345". Below it, a red error message reads: "Password are different!".
- Profile Image:** A placeholder box with the text "Please pick the image".
- Buttons:** Three red buttons are located at the bottom: "PICK IMAGE", "Register", and "Switch to Sign up".



### 3. Błędny format pliku wgrywanego jako zdjęcie

**Warunek wstępny:** Wgranie zdjęcia do profilu użytkownika.

**Oczekiwany rezultat:** Komunikat o błędnym formacie zdjęcia.

**Warunki końcowe:** Przy błędnym formacie pliku (nie jpg, png, jpeg), wyświetla się ikona błędnego pliku. Brak tutaj informacji dla użytkownika o błędnym załadowaniu pliku. Przy poprawnym formacie, plik wgrywa się prawidłowo i zostaje dodany do bazy.

**Dane testowe:** Błędne: [plik o formacie \\*.docx](#), Poprawne: [plik o formacie \\*.png](#)

The image displays two versions of a user registration form side-by-side. Both forms contain the following fields and buttons:

- User Name:** Input field with the text 'oandres'.
- E-Mail:** Input field with the text 'o\_andres@poczta.wysi.edu.pl'.
- Password:** Input field with masked characters '\*\*\*\*\*'.
- Confirm Password:** Input field with masked characters '\*\*\*\*\*'.
- Profile Image:** A section with a preview image (a cat wearing glasses) on the left and a 'PICK IMAGE' button on the right.
- Buttons:** 'Register' and 'Switch to Sign up' buttons are located at the bottom of each form.

The left form shows the 'Profile Image' field with a cat image, while the right form shows a placeholder image.

Wgrany został plik z nieprawidłowym formatem

Plik w prawidłowym formacie

**Warunek wstępny:** Zarejestrowany użytkownik z dostępem do systemu

**Kroki reprodukcji:**

1. Wybranie opcji „Zaloguj się”
2. Wprowadzenie adresu e-mail
3. Wprowadzenie hasła

**Oczekiwany rezultat:** Zalogowanie do systemu.

**Warunki końcowe:** Zalogowanie do systemu, przejście do ekranu głównego użytkownika.

**Dane testowe:** [o\\_andres@ms.wysi.edu.pl](#), 12345

**Negatywne przypadki**

## 4.2 Poprawne logowanie do systemu

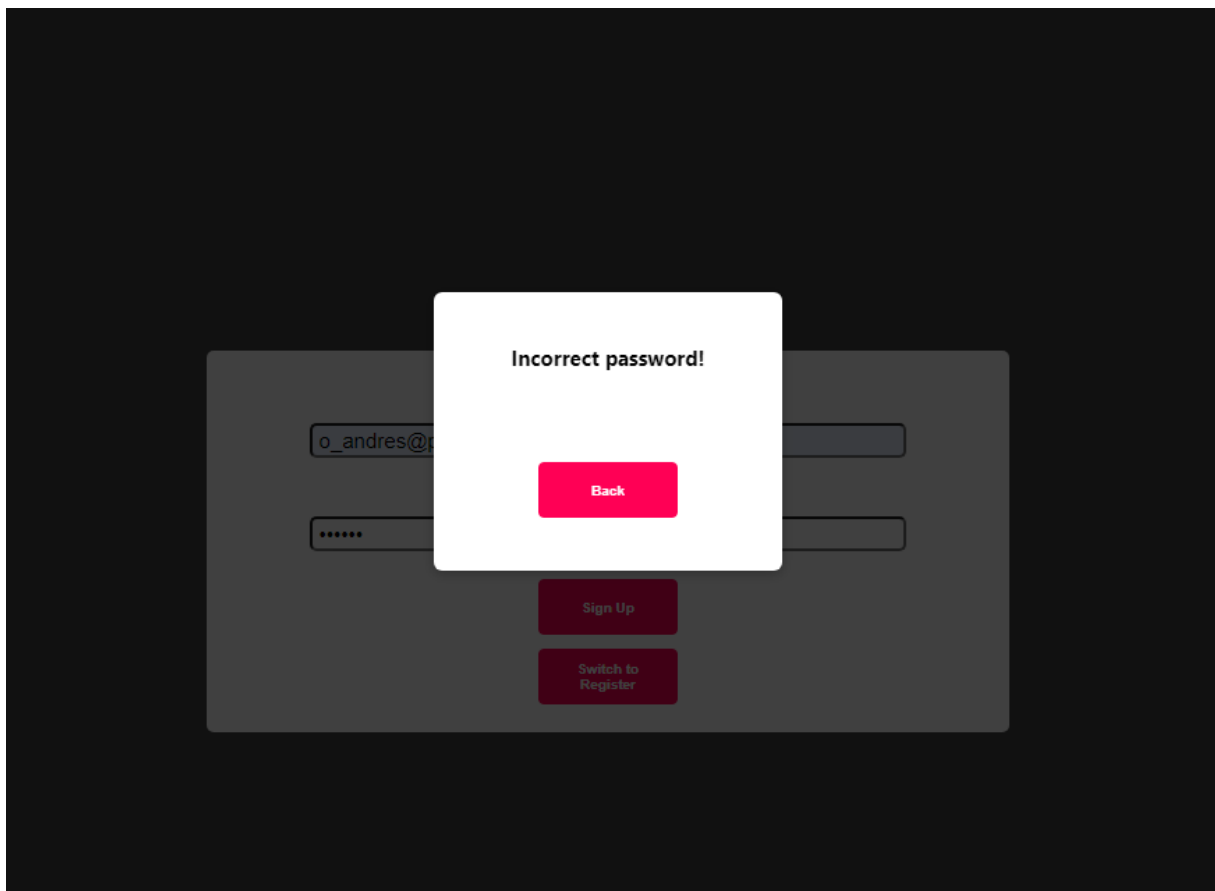
### 1. Błędne hasło

**Warunek wstępny:** Wpisanie błędnego hasła.

**Oczekiwany rezultat:** Brak możliwości logowania się do systemu.

**Warunki końcowe:** Przy błędnych danych w polu hasło otrzymujemy stosowny komunikat, oraz system nie przepuszcza użytkownika dalej.

**Dane testowe:** [o\\_andres@ms.wysi.edu.pl](mailto:o_andres@ms.wysi.edu.pl), 12346



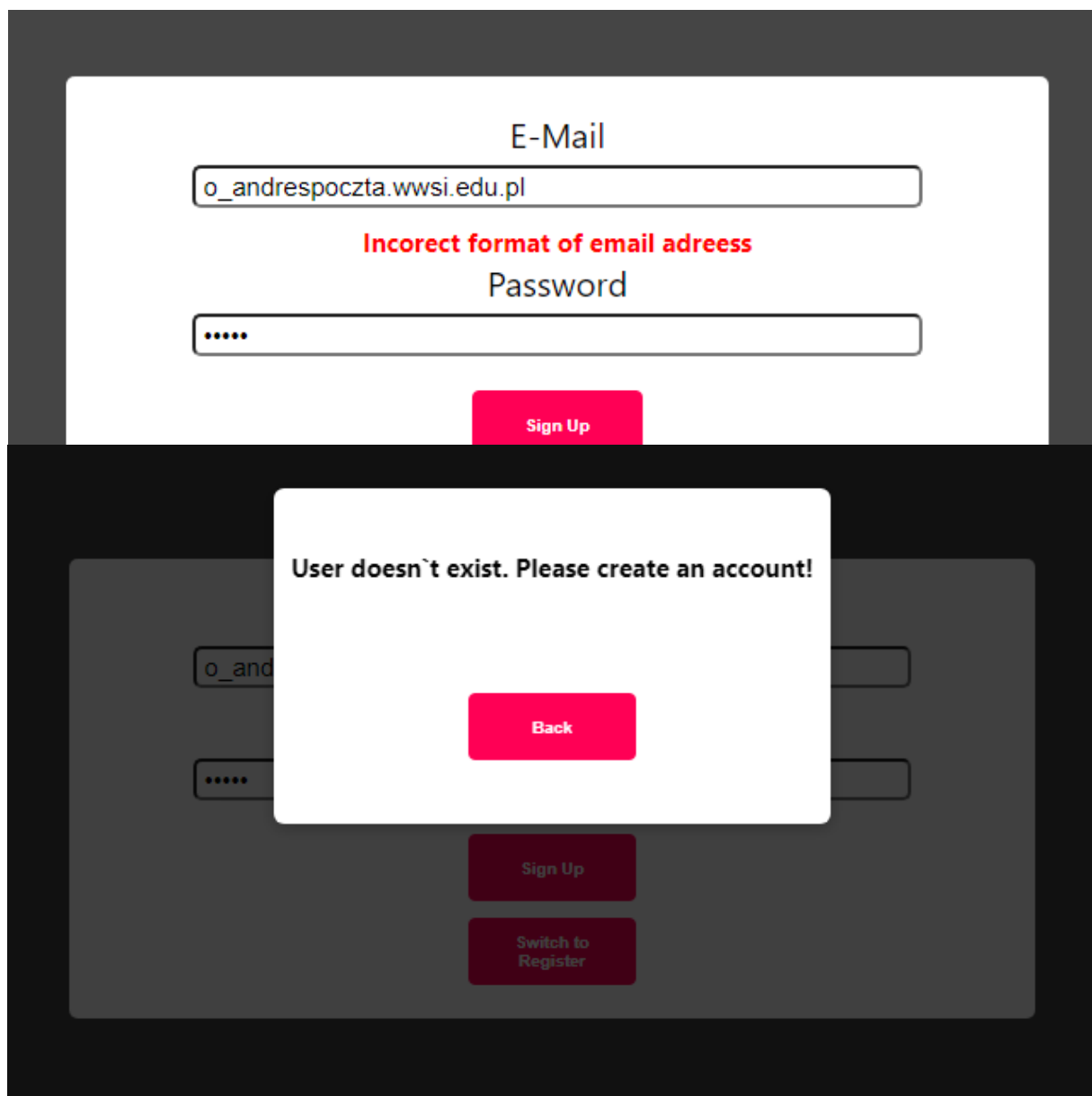
## 2. Błędny e-mail

**Warunek wstępny:** Wpisanie błędnego adresu e-mail.

**Oczekiwany rezultat:** Brak możliwości logowania się do systemu.

**Warunki końcowe:** Przy błędnych danych w polu e-mail otrzymujemy stosowny komunikat, oraz system nie przepuszcza użytkownika dalej.

**Dane testowe:** [o\\_andresms.wysi.edu.pl](mailto:o_andresms.wysi.edu.pl), [o\\_andres2@ms.wysi.edu.pl](mailto:o_andres2@ms.wysi.edu.pl), 12345



## 4.3 Sprawdzanie dodanych miejsc

**Warunek wstępny:** Wejście na podstronę z dodanymi przez nas miejscami.

**Oczekiwany rezultat:** Widok dodanych przez nas miejsc.

**Warunki końcowe:** Po wejściu na stronę z dodanymi przez nas miejscami widzimy listę naszych miejsc. Jeśli miejsce nie zostało przez nas dodane, nie wyświetla się nic oprócz komunikatu sugerującego dodanie miejsca.

**Negatywne przypadki:**

Brak dodanych miejsc przekierowuje nas na monit o konieczności dodania nowego miejsca. Niestety przycisk przekierowujący nas na formularz dodania nowego miejsca **nie działa**.

## 4.4 Dodawanie nowych miejsc

**Warunek wstępny:** Wejście na podstronę z formularzem dodawania miejsc. Wypełnienie poprawnie formularza.

**Oczekiwany rezultat:** Dodanie naszego miejsca do systemu. Widok nowego miejsca na podstronie z dodanymi miejscami.

**Warunki końcowe:** Po uzupełnieniu formularza poprawnymi danymi dostajemy informacje o poprawności wykonanego zadania.

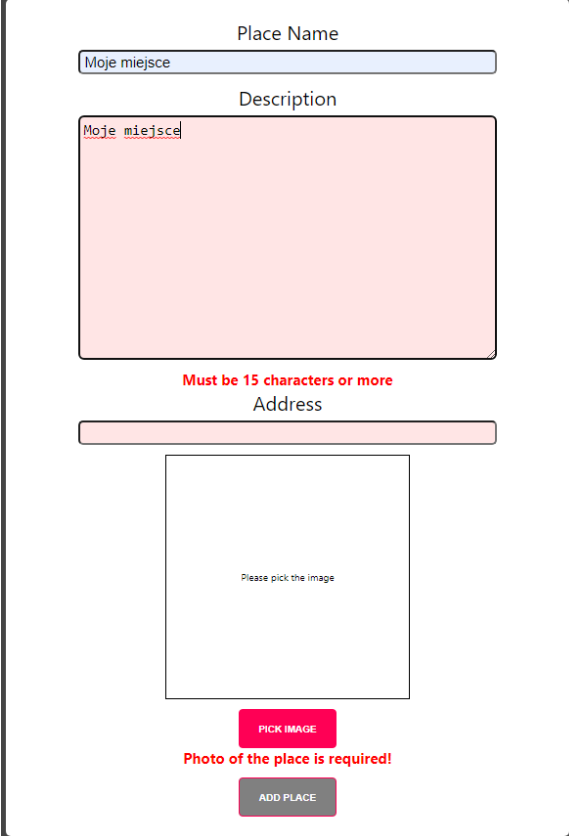
**Negatywne przypadki:**

## 1. Zbyt krótki opis miejsca

**Warunek wstępny:** Wpisanie zbyt krótkiego opisu miejsca.

**Oczekiwany rezultat:** Brak możliwości dodania obiektu, informacja dla użytkownika.

**Warunki końcowe:** Po wpisaniu zbyt krótkiego opisu miejsca dostajemy powiadomienie.



The image shows a web form for adding a place. It contains the following elements:

- Place Name:** A text input field with the placeholder text "Moje miejsce".
- Description:** A large text area with the placeholder text "Moje miejsce". Below it, a red error message states: "Must be 15 characters or more".
- Address:** A text input field.
- Image:** A square placeholder box with the text "Please pick the image".
- Buttons:** A red button labeled "PICK IMAGE" and a grey button labeled "ADD PLACE".
- Message:** A red error message below the image placeholder: "Photo of the place is required!".

## 2. Brak adresu

**Warunek wstępny:** Nie wpisanie adresu obiektu.

**Oczekiwany rezultat:** Brak możliwości dodania obiektu, informacja dla użytkownika.

**Warunki końcowe:** Bez wpisania adresu obiektu nie możemy go dodać do systemu.

### 3. Brak zdjęcia, zdjęcie w złym formacie

**Warunek wstępny:** Brak wybranego zdjęcia.

**Oczekiwany rezultat:** Brak możliwości dodania obiektu, informacja dla użytkownika.

**Warunki końcowe:** Przy błędnym formacie pliku (nie jpg, png, jpeg), wyświetla się ikona błędnego pliku. Brak tutaj informacji dla użytkownika o błędnym załadowaniu pliku. Przy poprawnym formacie, plik wgrywa się prawidłowo i zostaje dodany do bazy.

The form is titled 'Place Name' and contains a text input field with the placeholder text 'Moje miejsce'. Below this is a 'Description' section with a text area containing the text 'Moje miejsce pod miastem'. The 'Address' section has a text input field. Below the address field is a red 'Required' label. Underneath is a large square placeholder for an image with the text 'Please pick the image'. At the bottom of the form are two buttons: a red 'PICK IMAGE' button and a grey 'ADD PLACE' button. A red error message 'Photo of the place is required!' is displayed between the two buttons.

Place Name

Moje miejsce

Description

Moje miejsce pod miastem

Address

Required

Please pick the image

PICK IMAGE

Photo of the place is required!

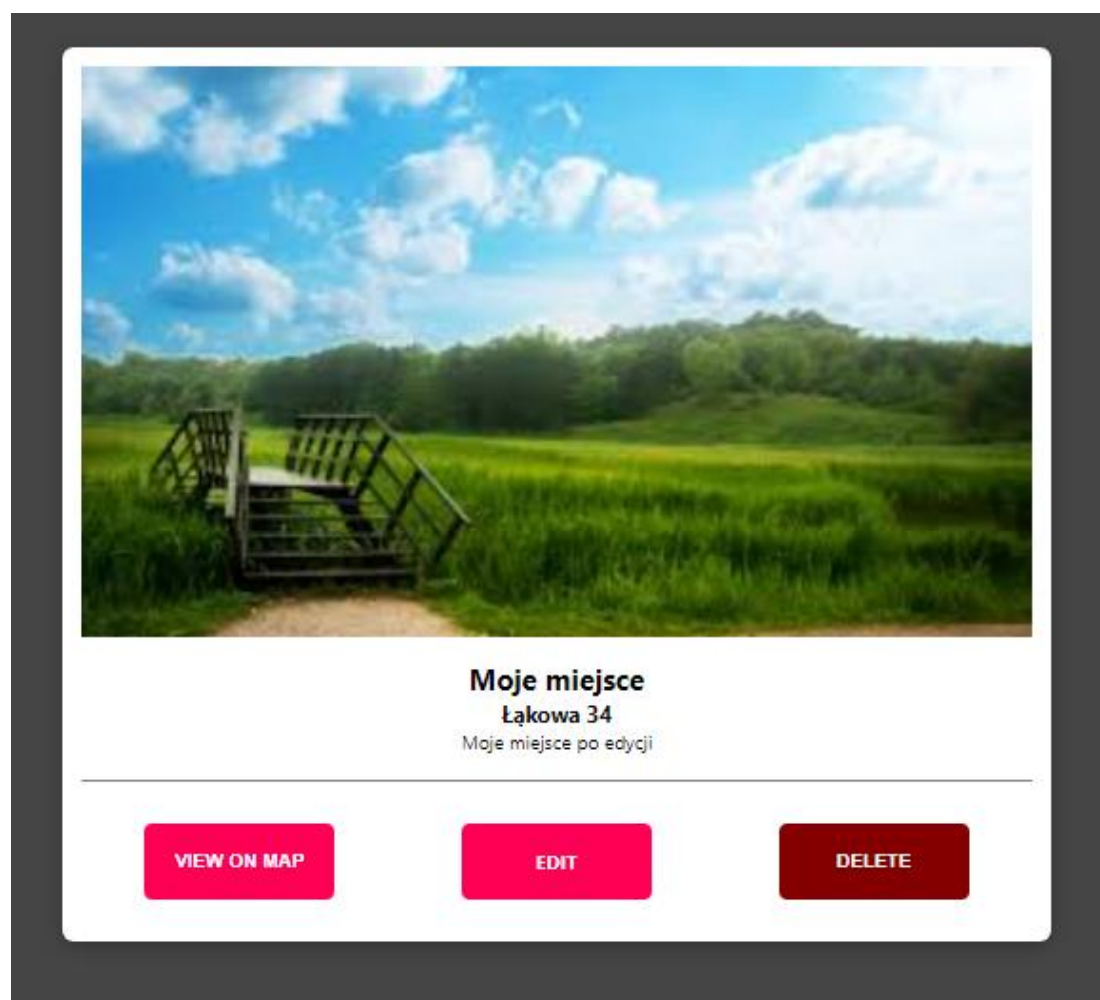
ADD PLACE

## 4.4 Edycja dodanego miejsca

**Warunek wstępny:** Wejście na podstronę z listą miejsc, wybranie edycji dodanego już wcześniej miejsca.

**Oczekiwany rezultat:** Edycja danych naszego miejsca.

**Warunki końcowe:** Po edycji danych, informacje o naszym miejscu powinny się zmienić.



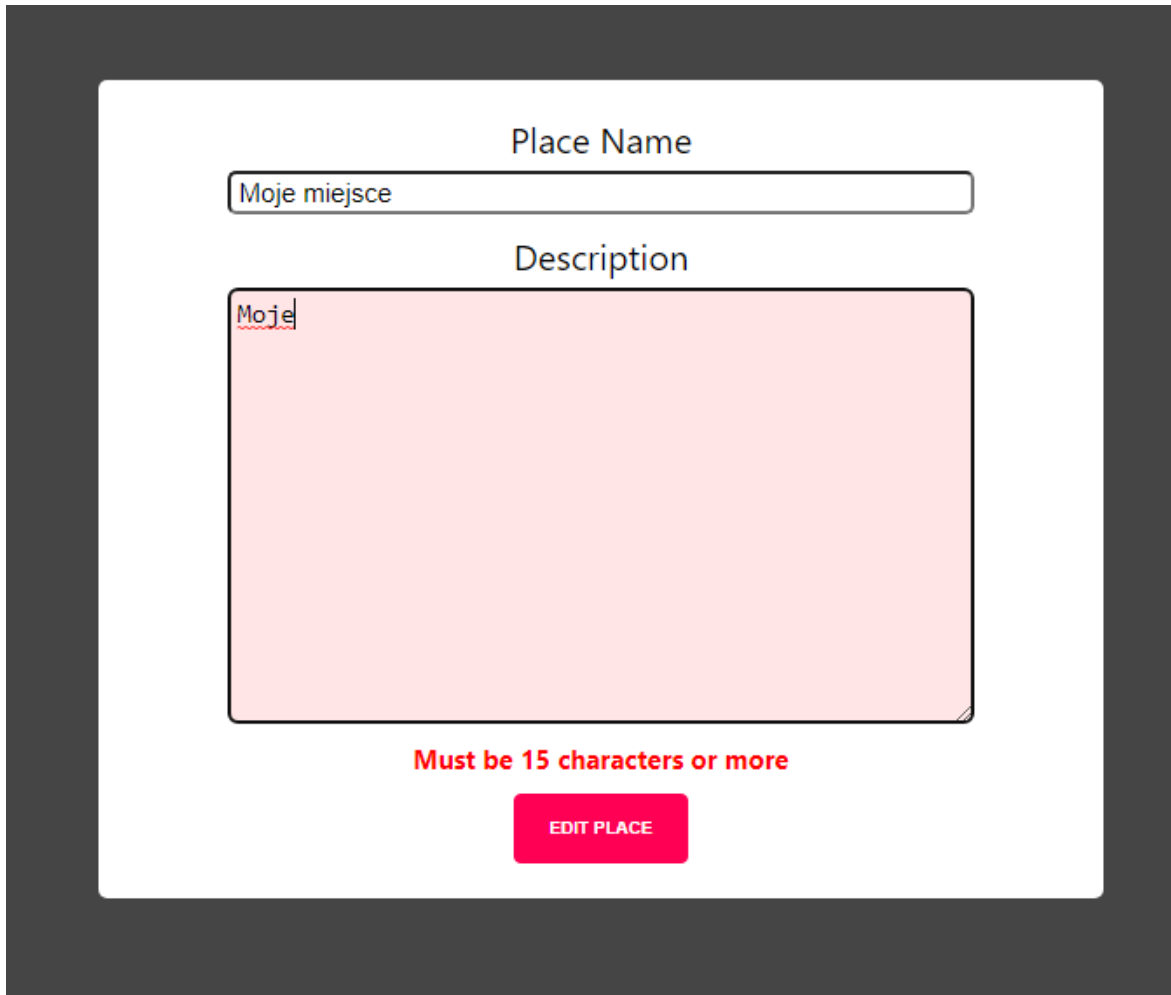
**Negatywne przypadki:**

## 1. Zbyt krótki opis miejsca

**Warunek wstępny:** Wpisanie zbyt krótkiego opisu miejsca.

**Oczekiwany rezultat:** Brak możliwości dodania obiektu, informacja dla użytkownika.

**Warunki końcowe:** Po wpisaniu zbyt krótkiego opisu miejsca dostajemy powiadomienie.



Place Name

Moje miejsce

Description

Moje

Must be 15 characters or more

EDIT PLACE

## 4.5 Usuwanie dodanego miejsca

**Warunek wstępny:** Posiadanie dodanego miejsca w systemie.

**Oczekiwany rezultat:** Usunięcie miejsca z systemu przez użytkownika.

**Warunki końcowe:** Po wybraniu Usuń, całkowite usunięcie miejsca z systemu.

**Negatywne przypadki:**



## 1. Przypadkowe naciśnięcie przycisku usuń

**Warunek wstępny:** Posiadanie dodanego miejsca w systemie, wybranie przycisku „Usuń”

**Oczekiwany rezultat:** Komunikat potwierdzający usunięcie.

**Warunki końcowe:** Po wybraniu Usuń dostajemy powiadomienie potwierdzające, dopiero po potwierdzeniu chęci usunięcia usuwamy miejsce z systemu.

## 4.6 Wylogowanie

**Warunek wstępny:** Posiadanie konta użytkownika, zalogowanie do systemu.

**Oczekiwany rezultat:** Po naciśnięciu przycisku „Wyloguj” natychmiastowe wylogowanie z systemu.

**Warunki końcowe:** Po wybraniu „Wyloguj” użytkownik zostaje natychmiastowo wylogowany i przeniesiony na stronę logowania. Widnieje również powiadomienie o udanej akcji.

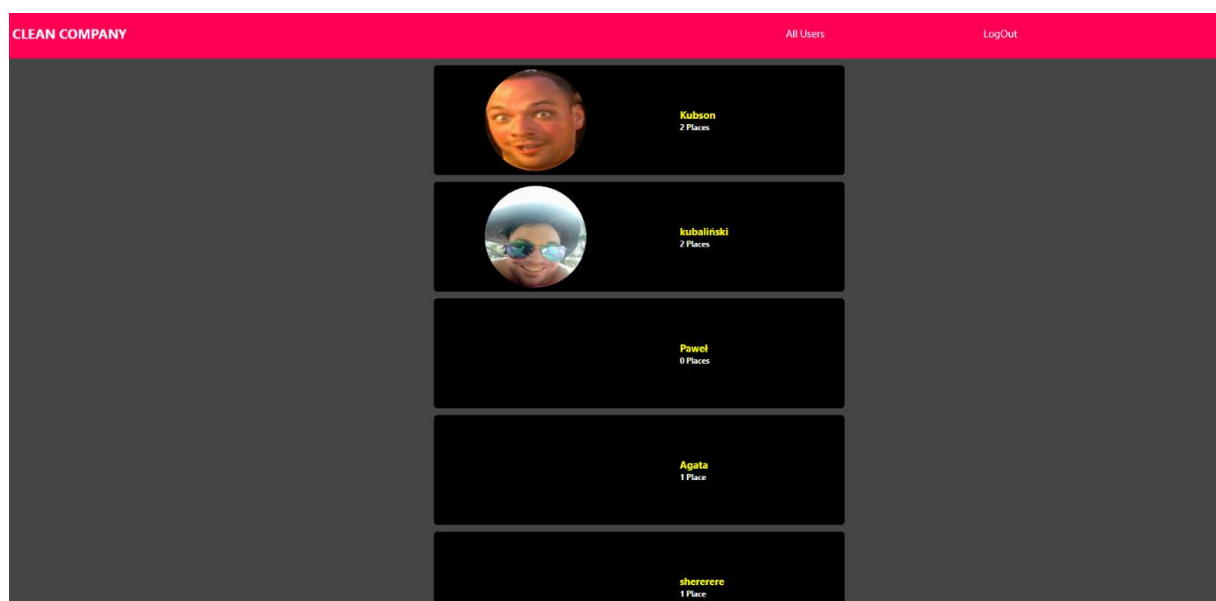
# PROFIL ADMINA

## 4.7 Sprawdzenie listy użytkowników

**Warunek wstępny:** Wejście na podstronę z widokiem wszystkich użytkowników

**Oczekiwany rezultat:** Widok użytkowników zarejestrowanych do systemu

**Warunki końcowe:** Po wejściu na stronę z dodanymi przez nas miejscami widzimy listę wszystkich użytkowników. Z tego miejsca można wykonać resztę procesów.



**Negatywne przypadki:**

### 1. Brak użytkowników w systemie

**Warunek wstępny:** Wejście na podstronę z widokiem wszystkich użytkowników.

**Oczekiwany rezultat:** Widok z brakiem użytkowników i informacją, że nie ma nikogo zalogowanego.

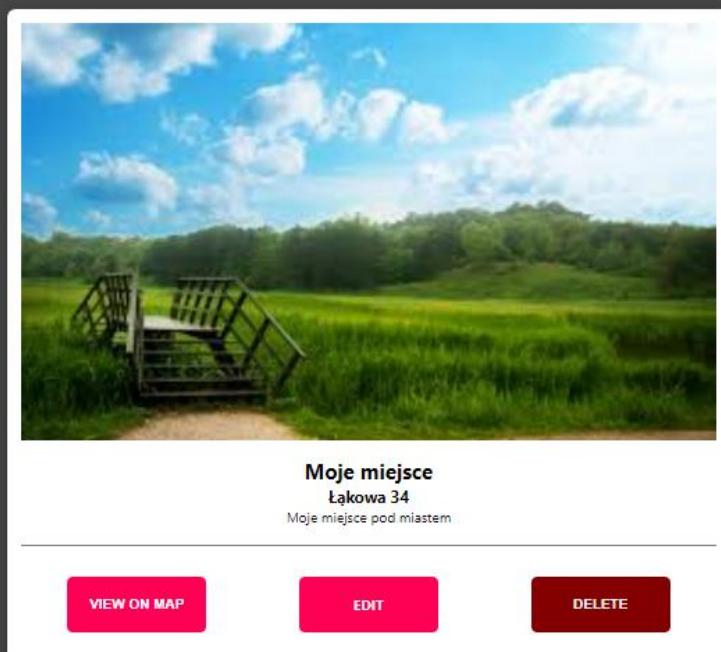
**Warunki końcowe:** Po wejściu na stronę z dodanymi przez nas użytkownikami nie widzimy ani jednego dodanego użytkownika, za to widzimy informację, że na chwilę obecną nikt się nie zarejestrował w naszym systemie.

## 4.8 Sprawdzenie profilu konkretnego użytkownika

**Warunek wstępny:** Wejście na podstronę z widokiem wszystkich użytkowników, wybranie użytkownika który ma profil w naszym systemie.

**Oczekiwany rezultat:** Widok profilu użytkownika, widok wszystkich dodanych przez niego miejsc.

**Warunki końcowe:** Po wejściu na profil wybranego użytkownika wyświetla nam się lista miejsc które dodał.



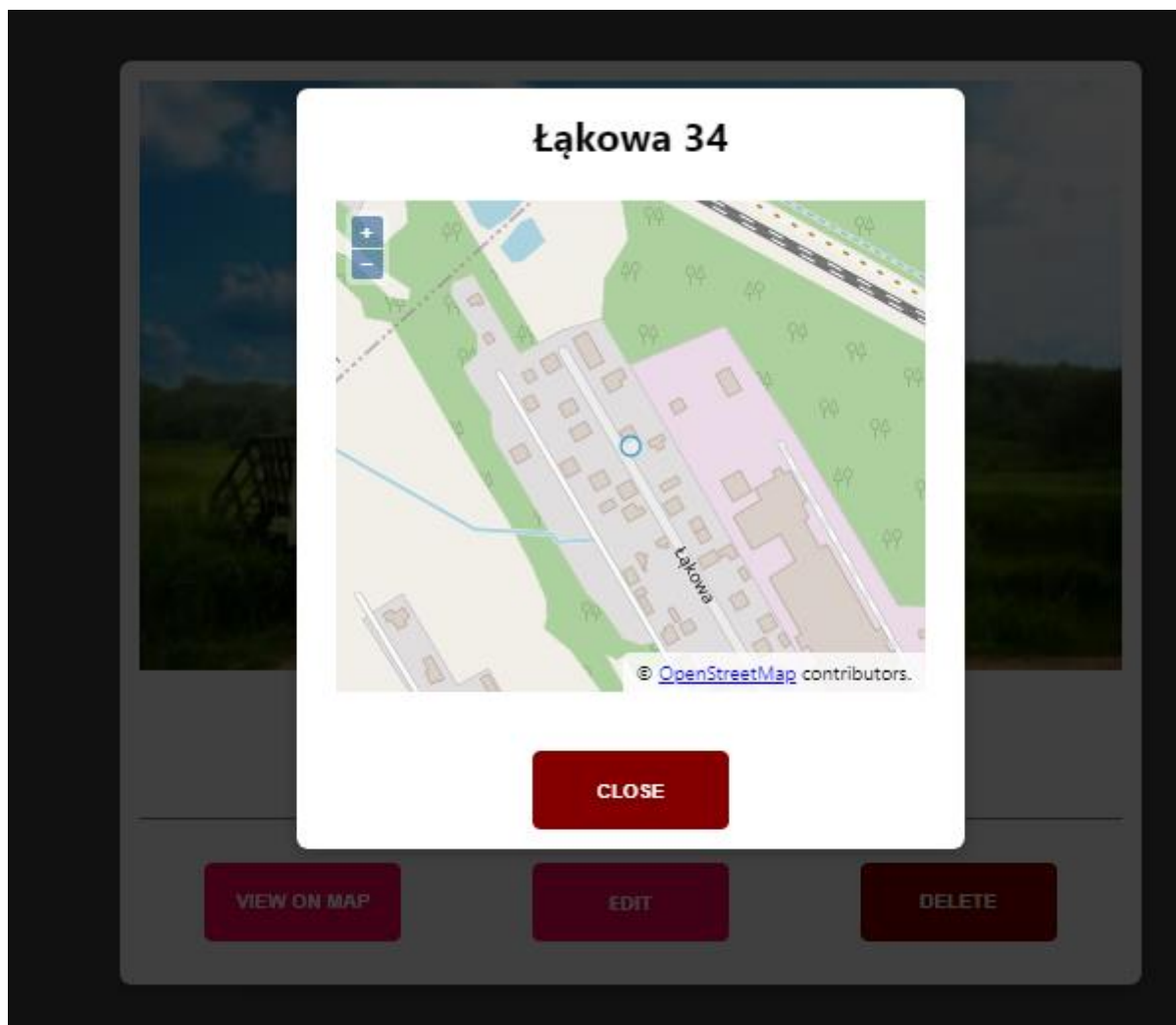
Admin ma możliwość edycji danych i sprawdzenia miejsca na mapie. Może również usunąć miejsce jeśli narusza ono politykę serwisu.

## 1. Sprawdzenie miejsca na mapie

**Warunek wstępny:** Posiadanie przez użytkownika poprawnie dodanego miejsca.

**Oczekiwany rezultat:** Widok dokładnej lokalizacji miejsca na mapie.

**Warunki końcowe:** Po kliknięciu w „Pokaż na mapie” wyświetla nam się mapa z dokładną lokalizacją miejsca.



## 2. Edycja danych przez admina

**Warunek wstępny:** Posiadanie przez użytkownika poprawnie dodanego miejsca.

**Oczekiwany rezultat:** Edycja danych miejsca, zaznaczenie, że lokal/miejsce zostało posprzątane

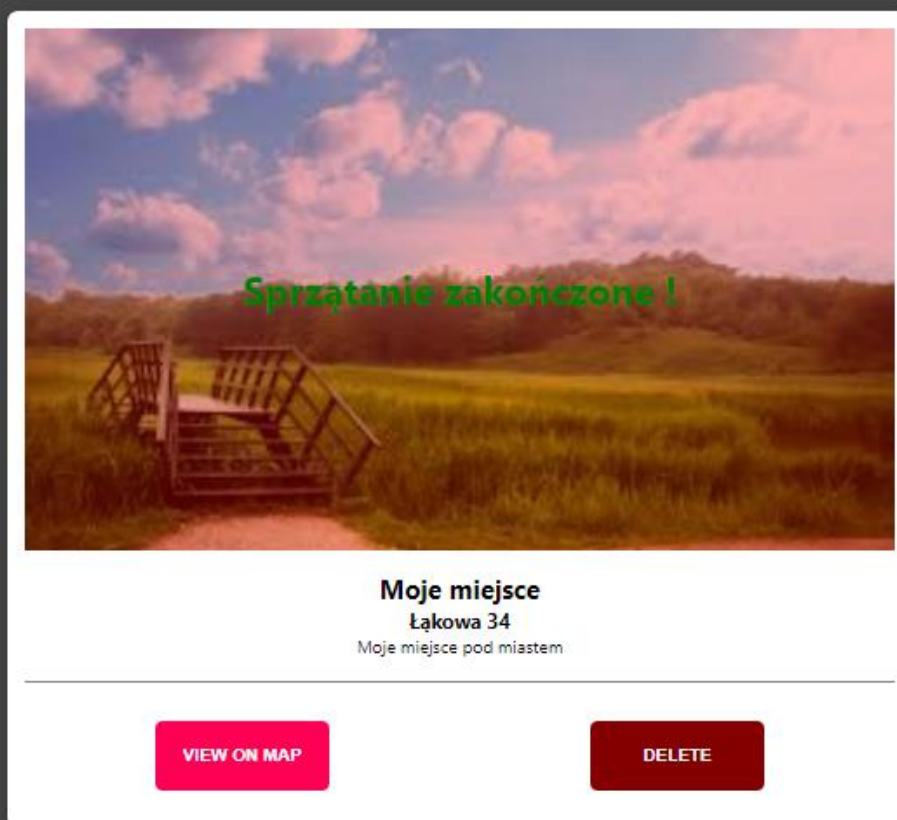
**Warunki końcowe:** Po kliknięciu w „Edytuj” przekierowuje nas na formularz edycji w którym admin ma możliwość edycji danych miejsca (opisu i nazwy) jak i również zaznaczenia czy lokal/miejsce zostało posprzątane.

## 3. Sprzątanie mieszkania

**Warunek wstępny:** Posiadanie przez użytkownika poprawnie dodanego miejsca które nie zostało posprzątane.

**Oczekiwany rezultat:** Po przestawieniu przycisku „Lokal czeka na sprzątanie”, lokal/miejsce zmienia swój status na posprzątane. Po zapisaniu edycji lokal zostaje całkowicie zamknięty i nie można dokonywać w nim edycji.

**Warunki końcowe:** Po kliknięciu przycisku „Lokal czeka na sprzątanie” zmienia się na napis „Loka posprzątany” jak i również lokal zmienia swój status. Po zapisaniu danych profil wygasa, zmienia się jego zdjęcie jak i również wyłączona jest możliwość edycji lokalu.



# WNIOSKI

Do poprawy zostaje usunięcie przycisku „Zaloguj się” z głównej strony tuż po zalogowaniu.

Po testach zostały znalezione liczne literówki które również trzeba poprawić.

Administratorzy systemu powinni trzymać się jednego formatu językowego.

Przycisk „Dodaj miejsce” po wejściu na pustą stronę z listą miejsc powinien przekierowywać na formularz z dodawaniem lokali.

Po sprzątnięciu lokalu, dobrym pomysłem byłoby dodanie przycisku który pozwala ponownie włączyć lokal (ponowne sprzątnięcie tego samego miejsca), zamiast tego użytkownik musi ponownie dodawać to samo miejsce.