**ETAP 4**

SYSTEM USŁUG SPRZĄTAJĄCYCH

|  |  |
| --- | --- |
| Imię i nazwisko studenta: | **Jakub Owczarzak, Andrzej Chmielewski**  **Rafał Spiżewski, Oliwia Andres, Michał Mackiewicz** |
| Grupa: | **Z710** |
| Specjalność: | **Inżynieria Internetu** |

Spis treści

[**1.** Implementacja Bazy danych 2](#_Toc93180834)

[1.1 Konfiguracja bazy danych 3](#_Toc93180835)

[1.2 Zarejestrowane zlecenia 3](#_Toc93180836)

[1.3 Użytkownicy 4](#_Toc93180837)

[**2.** **Implementacja warstwy logicznej** 5](#_Toc93180838)

[2.1 Logika biznesowa 5](#_Toc93180839)

[2.2 Architektura warstwowa 7](#_Toc93180840)

[**3.** **Implementacja GUI** 8](#_Toc93180841)

[3.1 Ekran główny dla osoby nie zalogowanej 9](#_Toc93180842)

[3.2 Ekran logowania 9](#_Toc93180843)

[3.3 Ekran rejestracji 9](#_Toc93180844)

[3.4 Ekran zalogowanego użytkownika 10](#_Toc93180845)

[3.5 Dodawanie miejsca wykonania usługi 10](#_Toc93180846)

[3.6 Widok dodanego przez użytkownika miejsca wykonania usługi 11](#_Toc93180847)

[3.7 Podgląd wybranego adresu na mapie 11](#_Toc93180848)

[3.8 Edycja miejsca wykonania usługi 12](#_Toc93180849)

[3.9 Usuwanie miejsca wykonania usługi 12](#_Toc93180850)

[3.10 Widok strony głównej dla konta Admin 13](#_Toc93180851)

[3.11 Widok zarejestrowanych użytkowników dla konta Admin 13](#_Toc93180852)

[3.12 Podgląd zgłoszonych miejsc dla wybranego użytkownika z poziomu konta Admin 14](#_Toc93180853)

[**4.** **Testy aplikacji** 14](#_Toc93180854)

[4.1 Poprawna rejestracja do systemu 14](#_Toc93180855)

[4.2 Poprawne logowanie do systemu 17](#_Toc93180856)

[4.3 Sprawdzanie dodanych miejsc 19](#_Toc93180857)

[4.4 Dodawanie nowych miejsc 20](#_Toc93180858)

[4.4 Edycja dodanego miejsca 22](#_Toc93180859)

[4.5 Usuwanie dodanego miejsca 24](#_Toc93180860)

[4.6 Wylogowanie 25](#_Toc93180861)

[4.7 Sprawdzenie listy użytkowników 25](#_Toc93180862)

[4.8 Sprawdzenie profilu konkretnego użytkownika 26](#_Toc93180863)

## Implementacja Bazy danych

### 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.

Obraz zawierający tekst

Opis wygenerowany automatycznie

### 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

Obraz zawierający tekst

Opis wygenerowany automatycznie

### 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

Obraz zawierający tekst

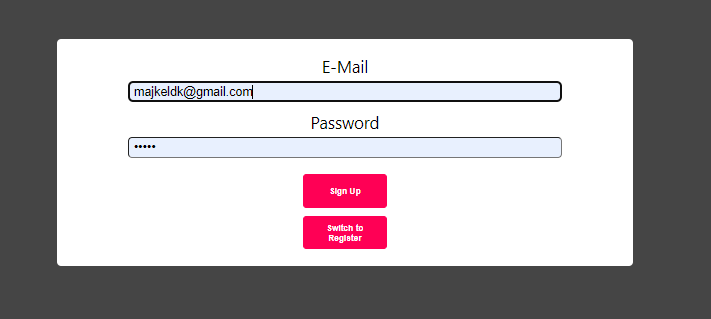
Opis wygenerowany automatycznie

## **Implementacja warstwy logicznej**

### Logika biznesowa

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

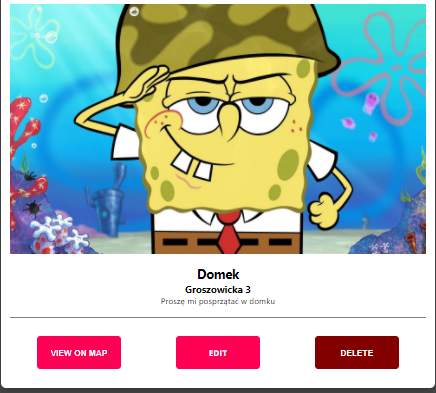
Użytkownik ma możliwość zalogowania do aplikacji:



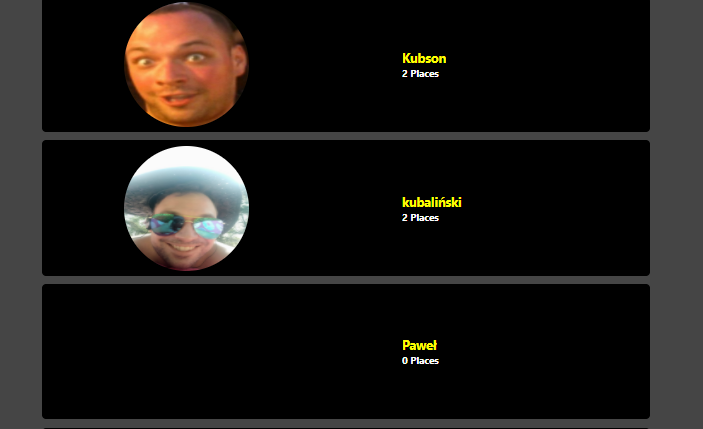
Dodania miejsca usługi

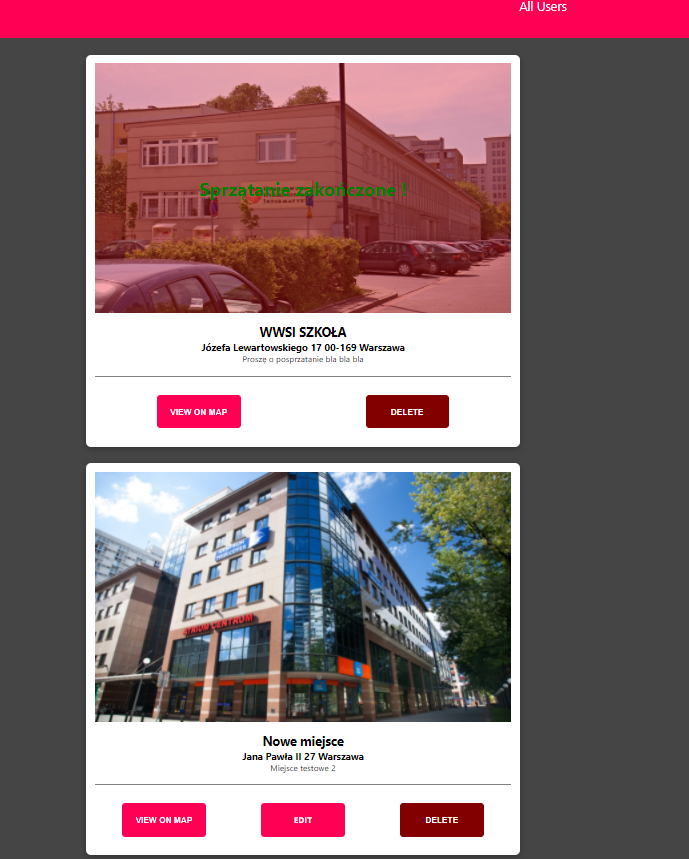
Podglądu dodanego miejsca wraz z możliwością jego edycji, podglądu na mapie i usunięcia

Obraz zawierający tekst

Opis wygenerowany automatycznie 

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



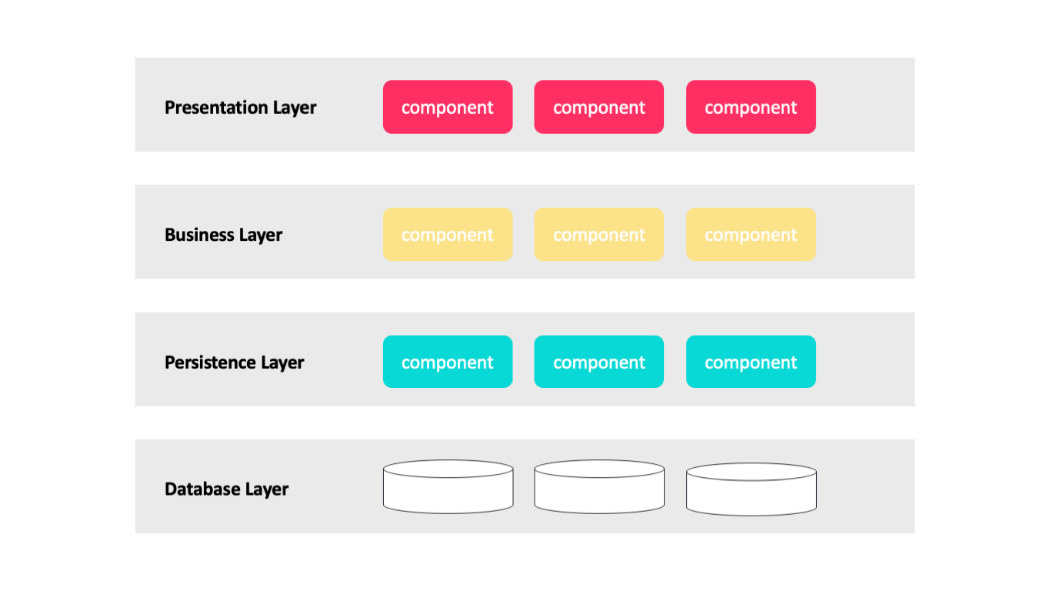


### Architektura warstwowa

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 napisanych 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 warstw 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 za internetowym interfejsem API.

Obraz zawierający tekst

Opis wygenerowany automatycznie

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

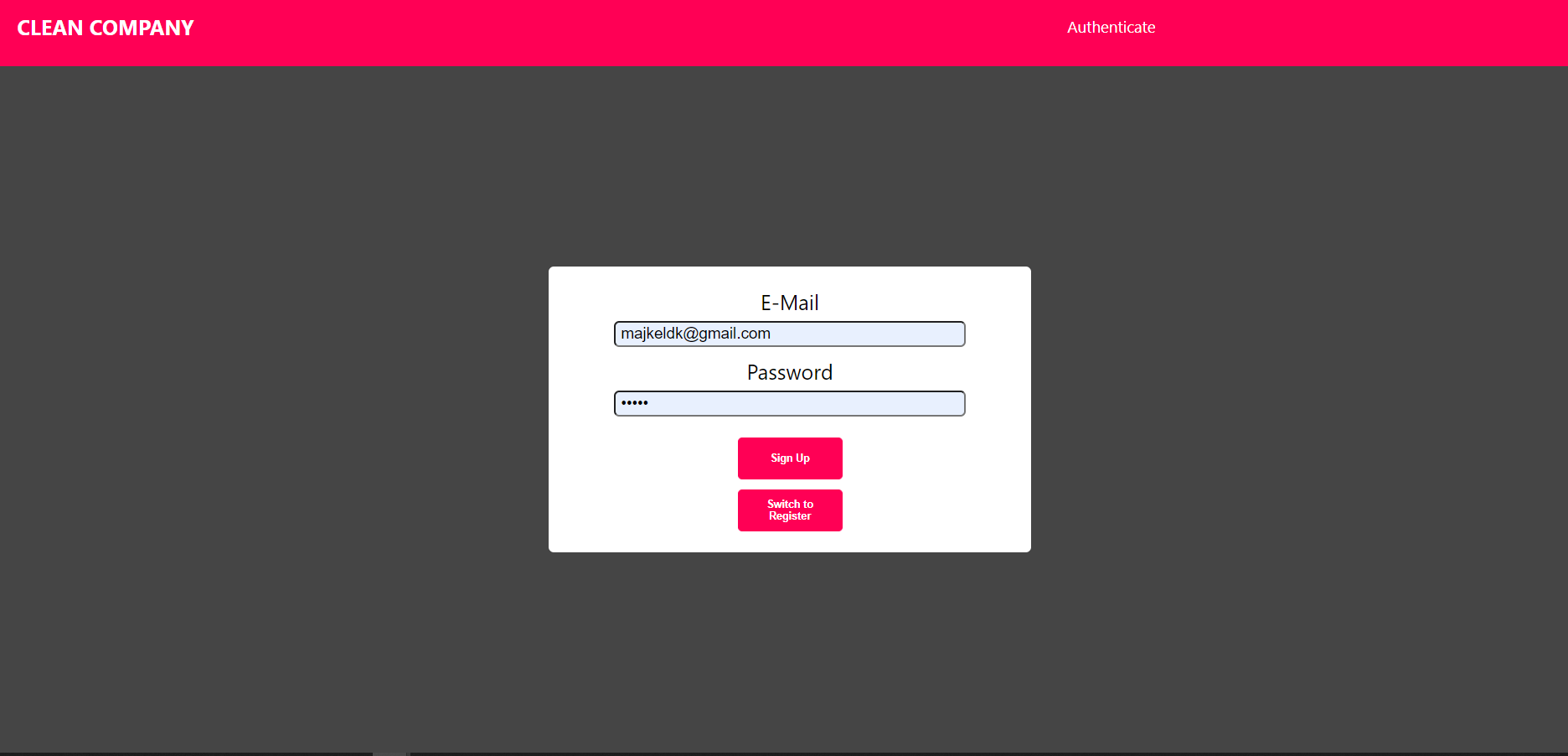
## **Implementacja GUI**

### Ekran główny dla osoby nie zalogowanej

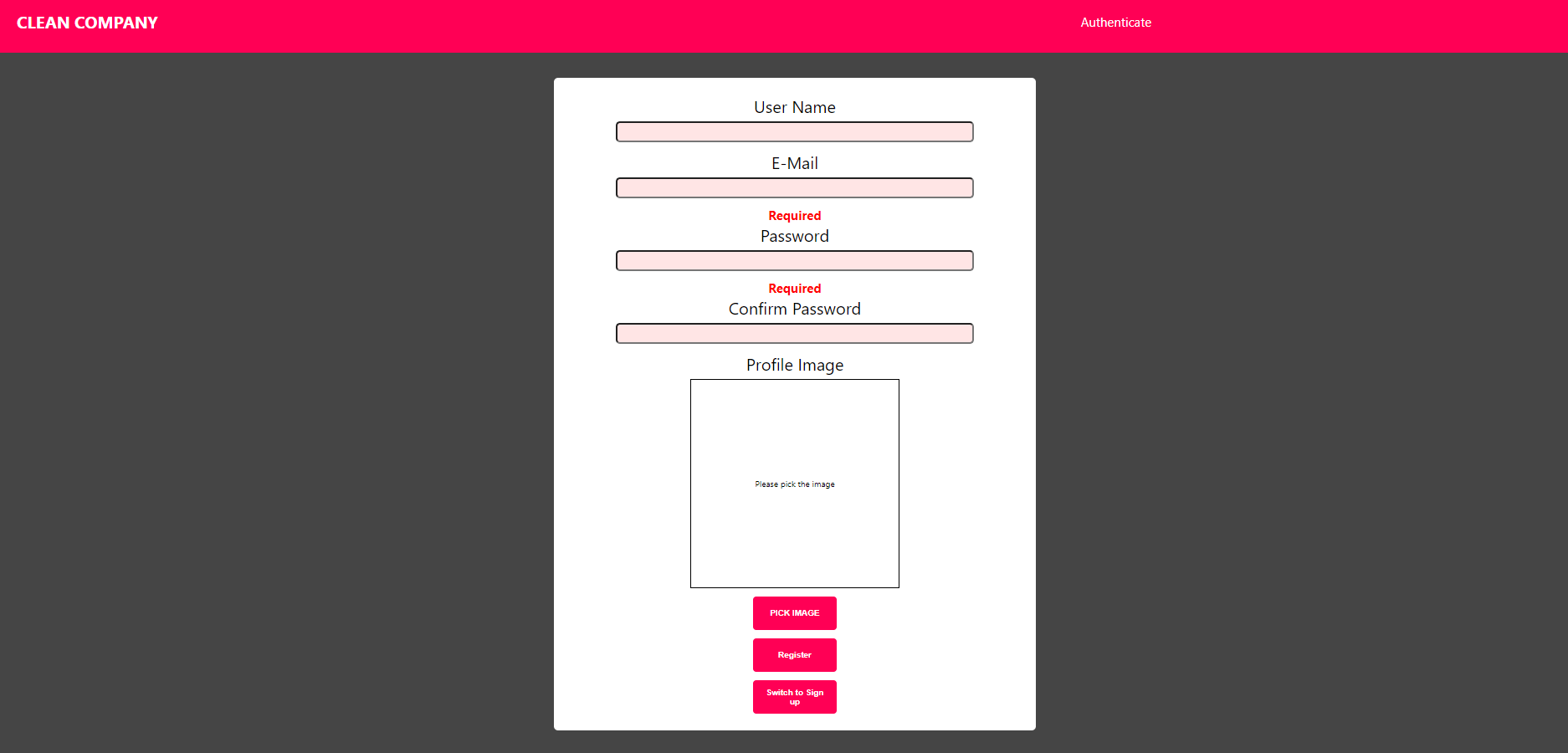
Obraz zawierający tekst

Opis wygenerowany automatycznie

### Ekran logowania



### Ekran rejestracji

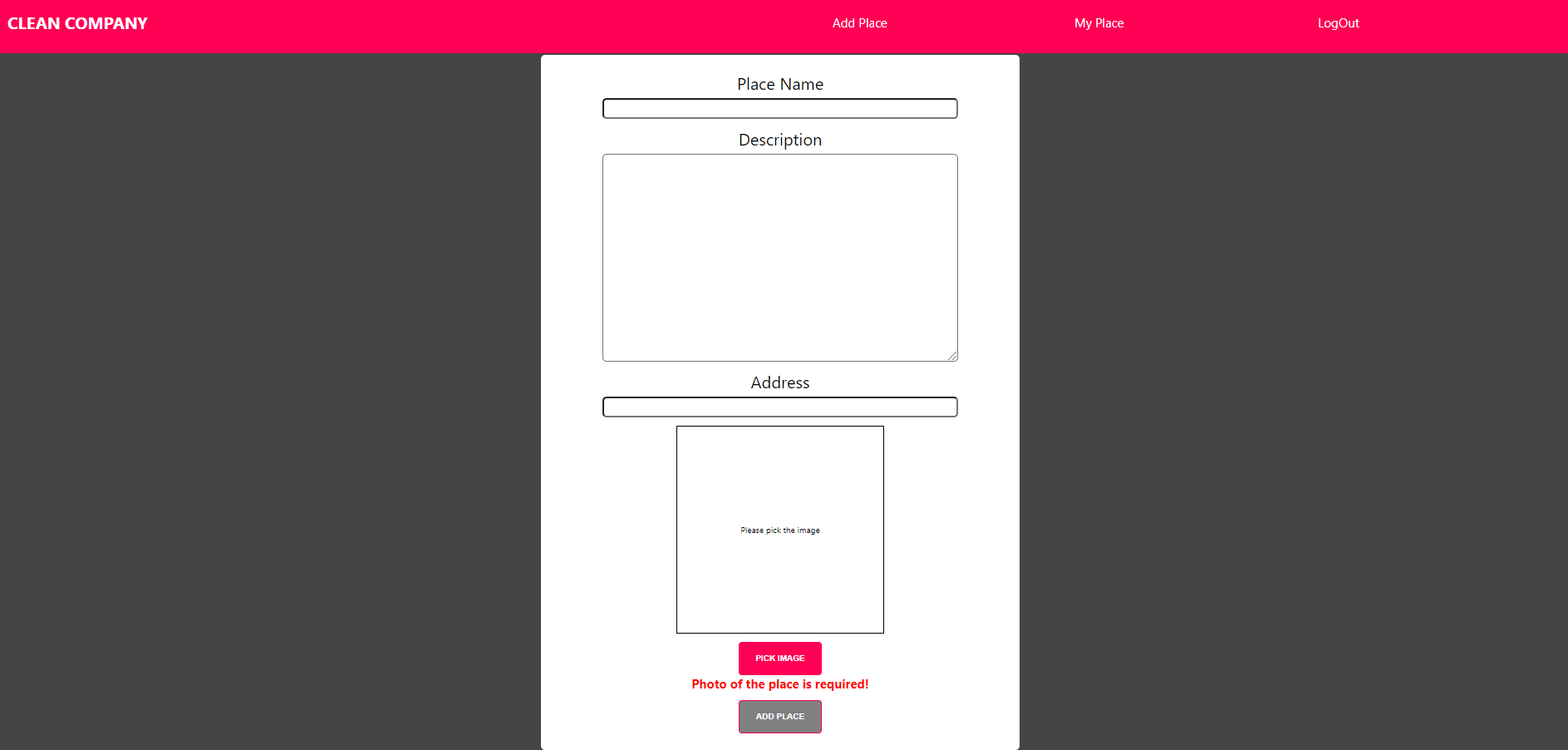


### Ekran zalogowanego użytkownika

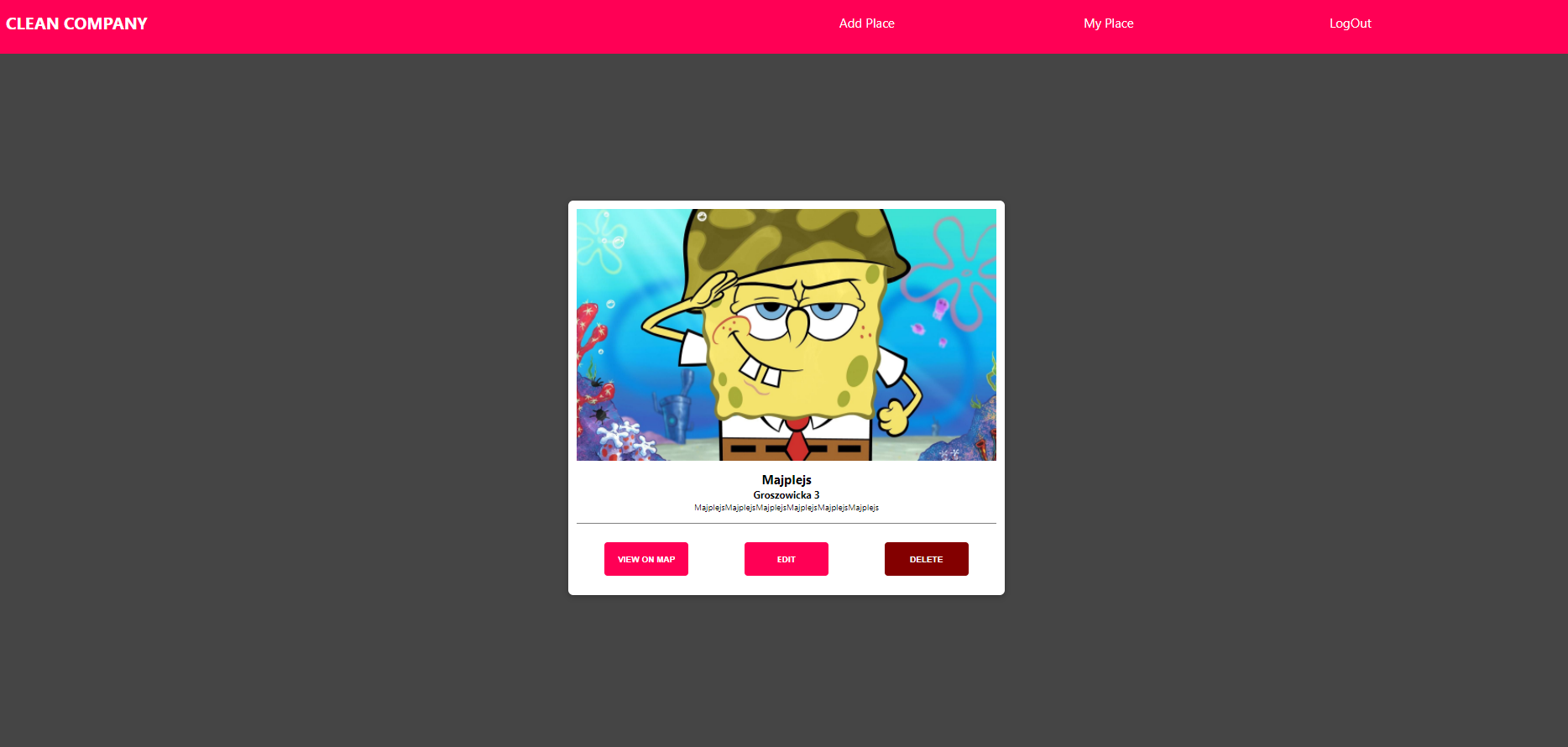
Obraz zawierający tekst

Opis wygenerowany automatycznie

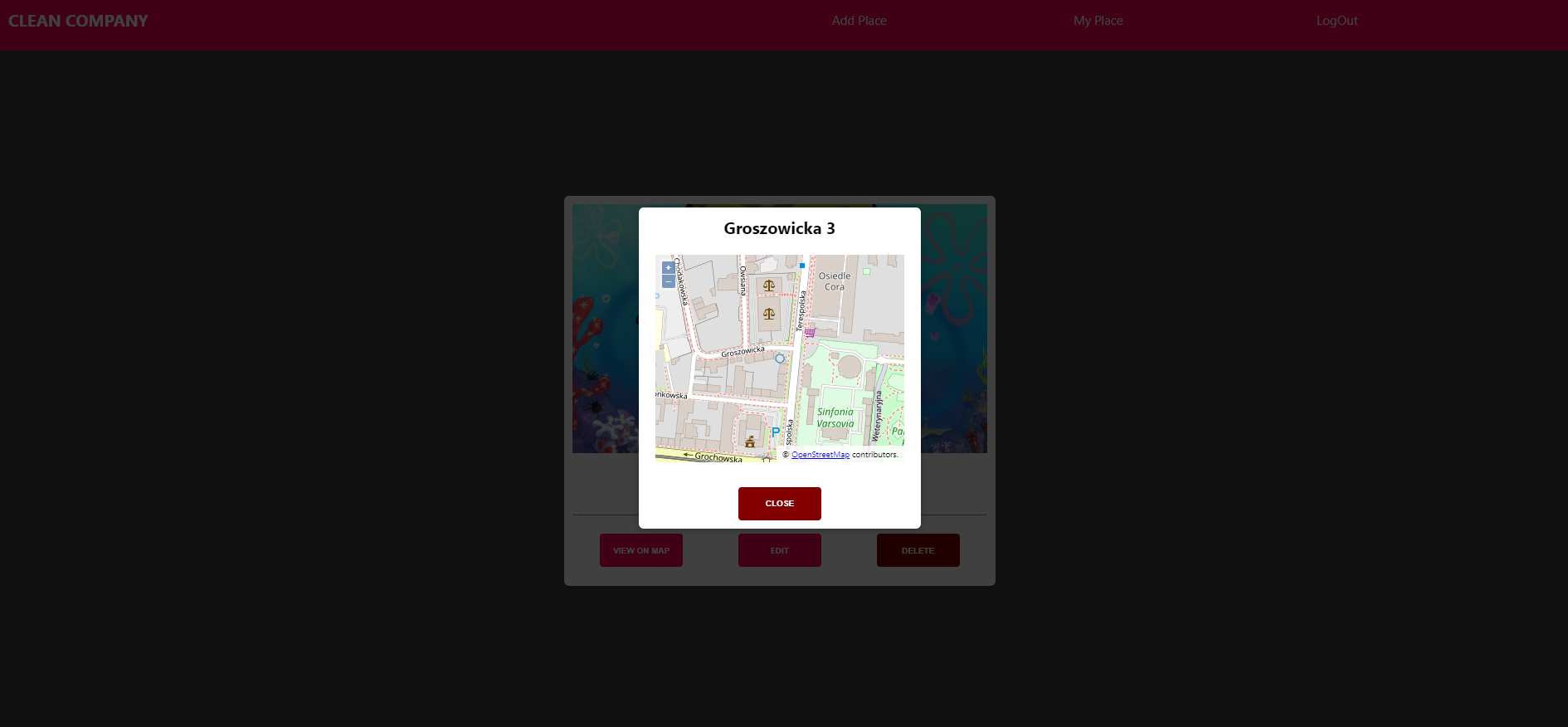
### Dodawanie miejsca wykonania usługi



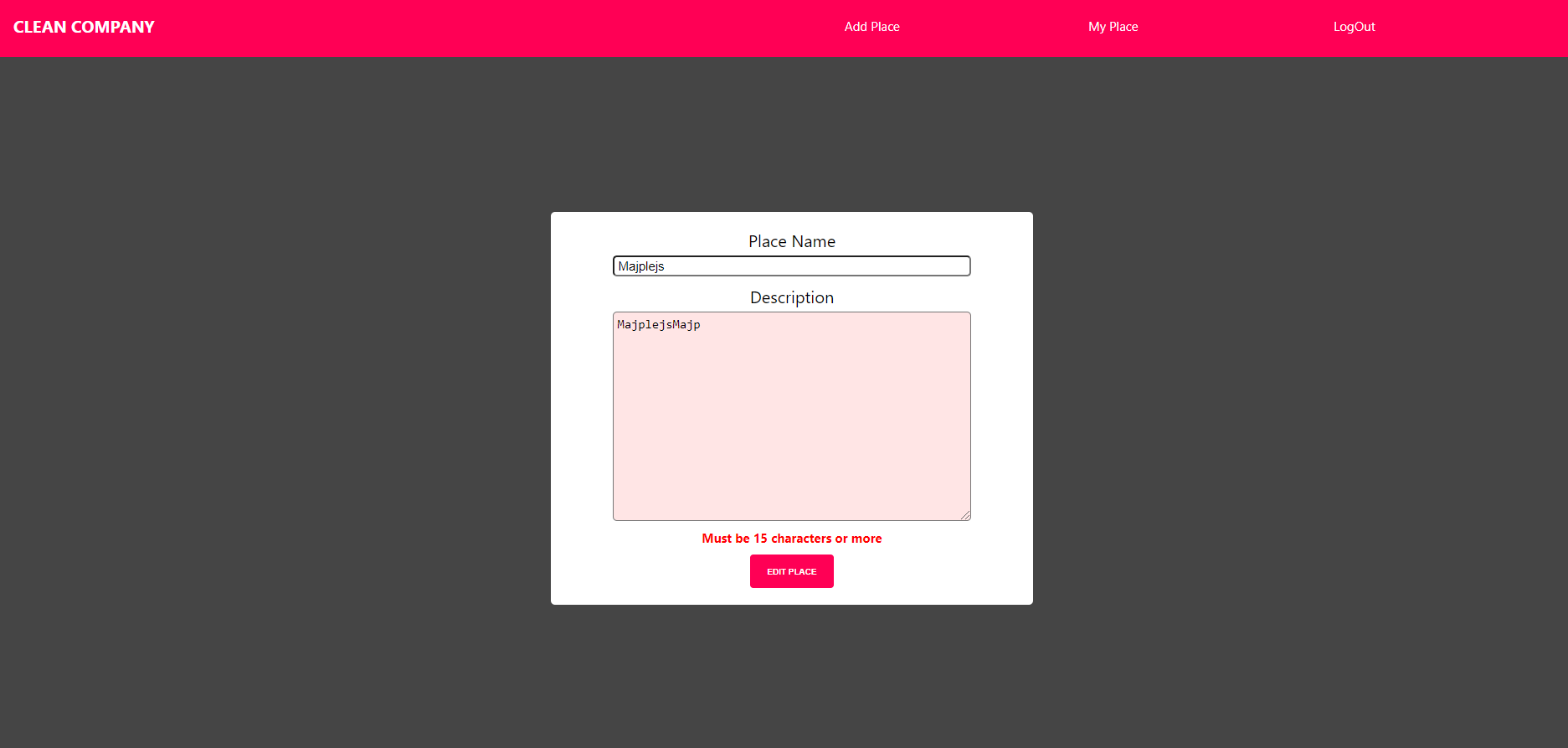
### Widok dodanego przez użytkownika miejsca wykonania usługi



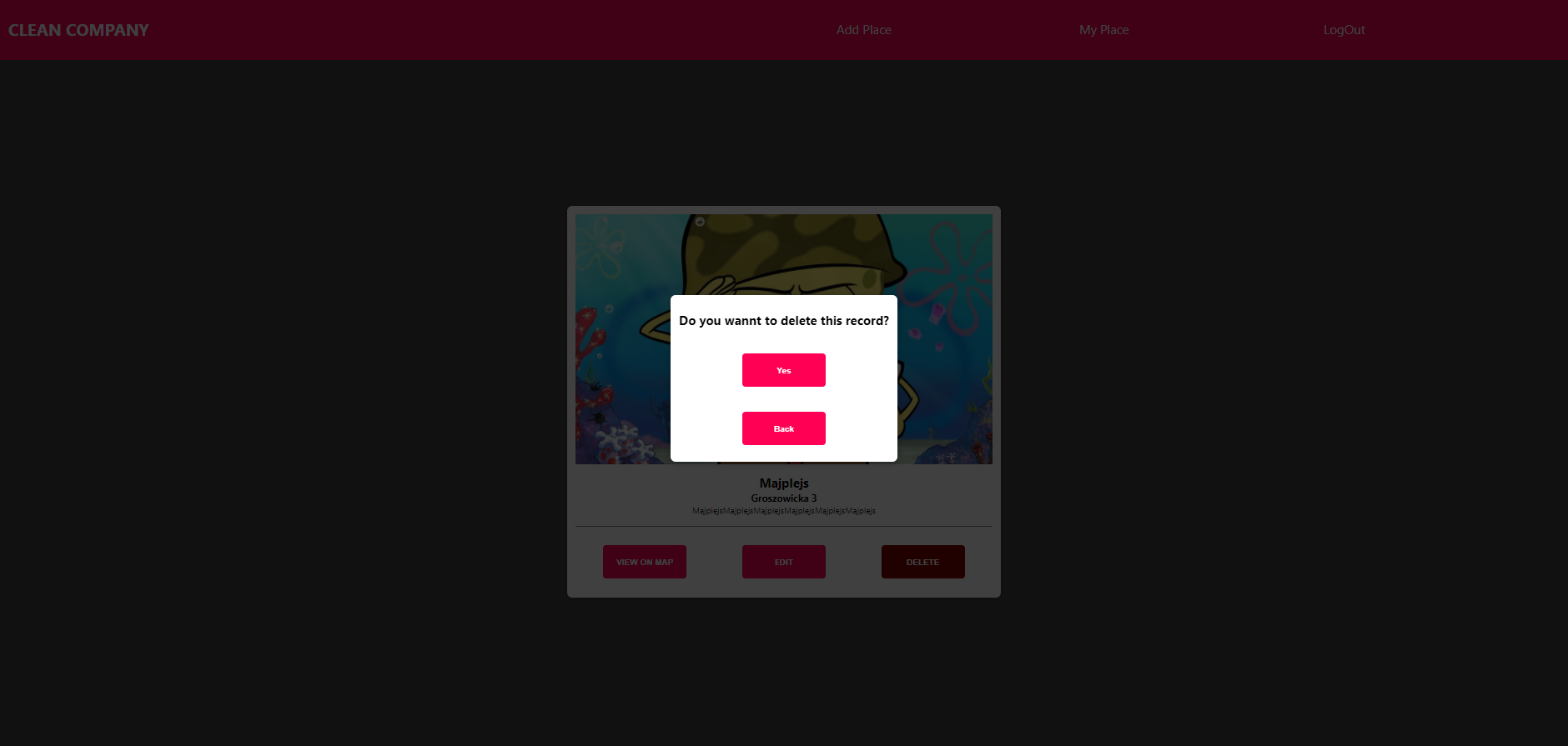
### Podgląd wybranego adresu na mapie



### Edycja miejsca wykonania usługi



### Usuwanie miejsca wykonania usługi

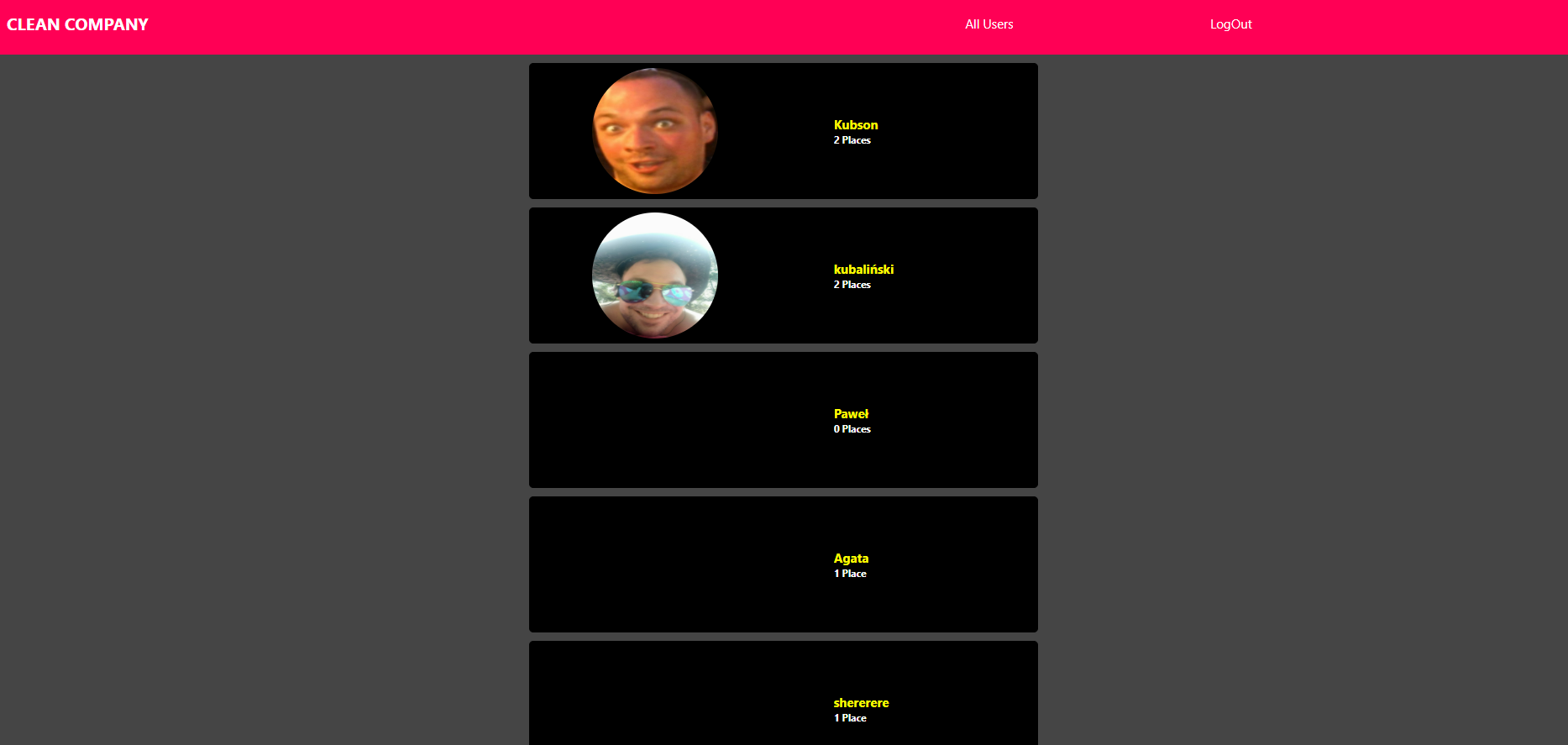


### Widok strony głównej dla konta Admin

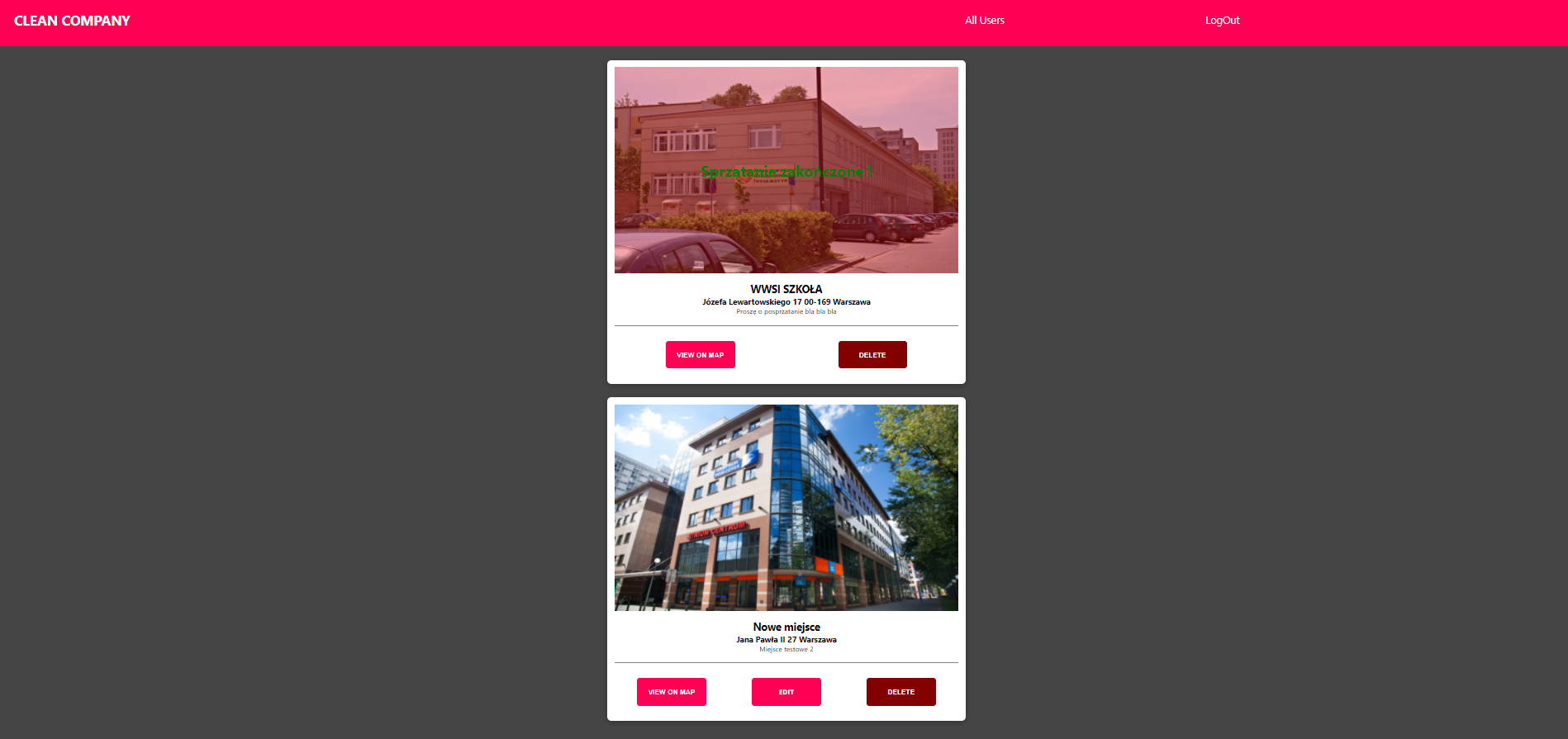
Obraz zawierający tekst

Opis wygenerowany automatycznie

### Widok zarejestrowanych użytkowników dla konta Admin



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



## **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.wwsi.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.

Graphical user interface

Description automatically generatedA picture containing graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated**Dane testowe**: Błędne*: o\_andresms.wwsi.edu.pl*, Poprawne: o\_andres@ms.wwsi.edu.pl

*Niepoprawne dane Poprawne dane*

**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

Graphical user interface

Description automatically generatedOdpowiednie komunikaty przy niepoprawnych

danych.

**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

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

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.wwsi.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.wwsi.edu.pl, 12346

Graphical user interface, diagram

Description automatically generated

**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.wwsi.edu.pl, o\_andres2@ms.wwsi.edu.pl, 12345

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

### 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.

Text

Description automatically generated with medium confidence

**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 formaci**e

**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.

A picture containing graphical user interface

Description automatically generated

### 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.

Graphical user interface, website

Description automatically generated

Negatywne przypadki:

**Negatywne przypadki**:

**1. Zbyt krótki opis miejsca**

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

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

Graphical user interface

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

### 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.

Graphical user interface

Description automatically generated

**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.

Graphical user interface

Description automatically generated

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

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.

Graphical user interface, website

Description automatically generated

**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.

Graphical user interface, website

Description automatically generated

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