# Technologie webowe w aplikacjach Internetu rzeczy II (Projekt)

## 1. Temat

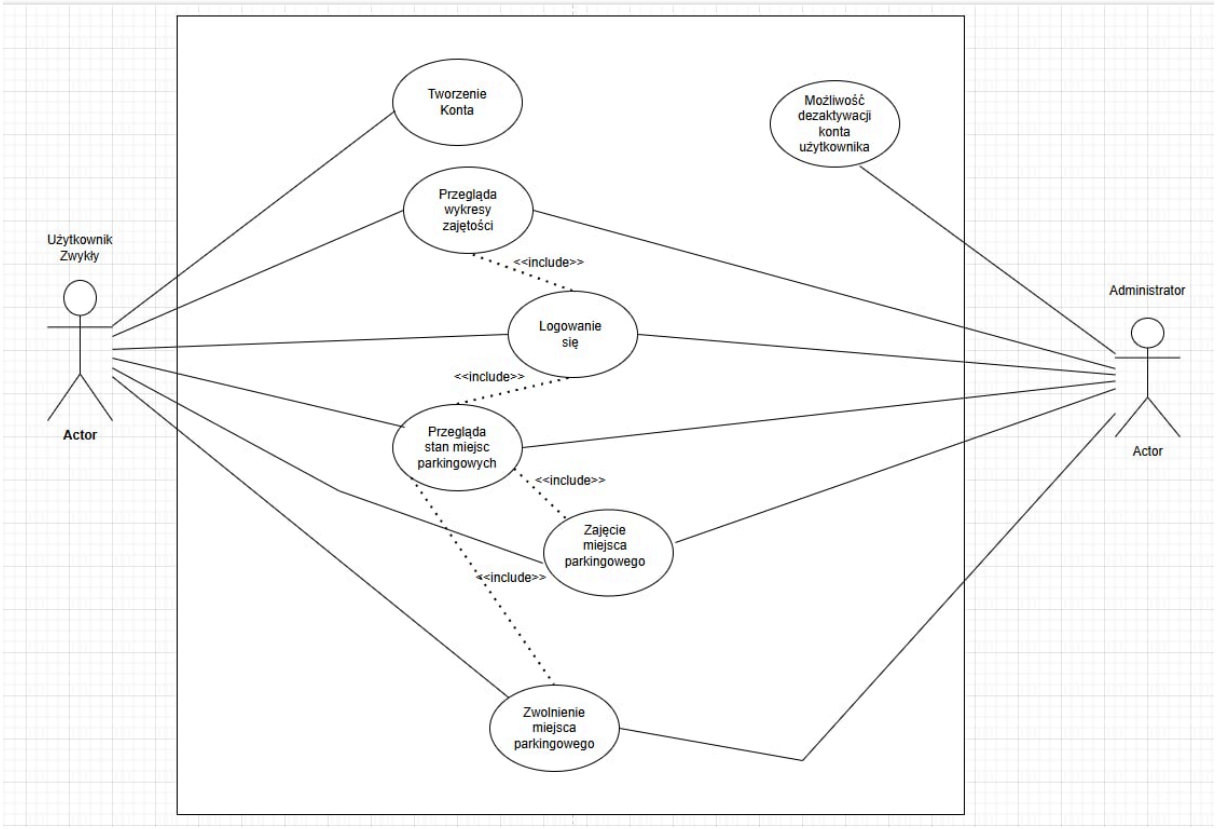
Inteligentny parking

## 2. Skład zespołu

Bartłomiej Podlewski [36403]
Dawid Osak [36400]

## 3. Opis

**Diagram przypadków użycia**




**Opis architektury:**

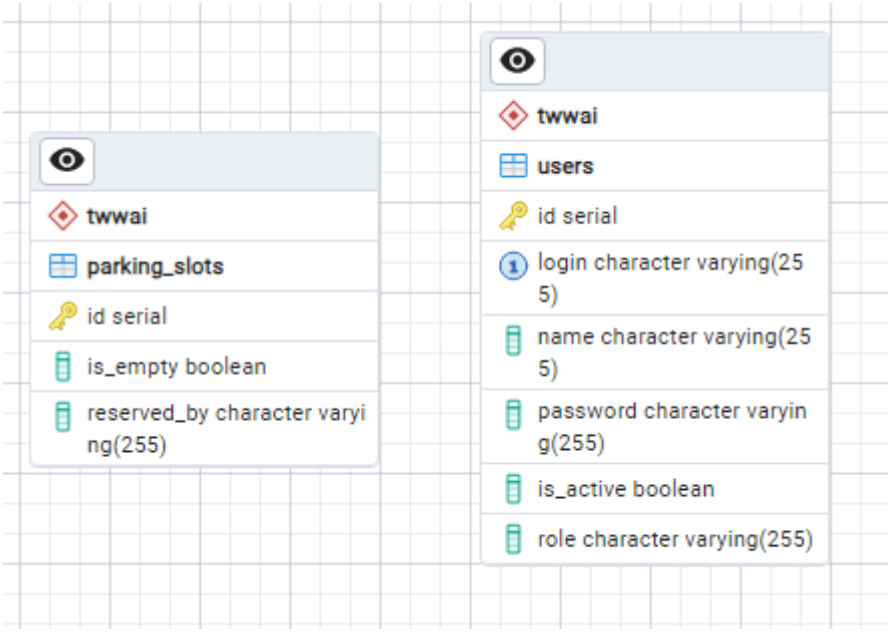
- Frontend (React)
  Z poziomu użytkownika jest możliwość zajmowania/zwalniania miejsc parkingowych. Użytkownik z rolą administratora może wygasić konta użytkowników. Komunikacja z backendem odbywa się przez HTTP.

- Backend (Java + Spring)
  Obsługuje żądania z frontendu, filtruje czy żądanie jest autoryzowane za pomocą JWT. Zarządza logiką biznesową i komunikuje się z bazą.

Controller – obsługa żądań http
Service – logika biznesowa
Repository – interakcja z bazą poprzez JPA/Hibernate

- Baza danych (PostgreSQL)

**Diagram ERD**




## Opis API (http://host:8080/api/v1)

@GET
/users/{login} – zwraca użytkownika  na podstawie loginu
/users/role – zwraca role użytkownika na podstawie tokenu
/users/all – zwraca listę użytkowników

@POST
/auth/register – wprowadzanie użytkownika do bazy, zwraca token

```
{
    "name": "…",
    "login": "…",
    "password": "…"
}
```

/auth/authenticate – autentykacja użytkownika, zwraca token

```
{
    "login": "…",
    "password": "…"
}
```

/parking-slots/create – tworzy miejsce parkingowe w bazie

@PUT
/users/{login}/toggleActive – zmiana stanu aktywności konta użytkownika
/parking-slots/{id}/occupy – zajmuje wybrane miejsce parkingowe
/parking-slots/occupy – zajmuje pierwsze wolne miejsce
/parking-slots/release – zwalnia miejsce zajęte przez użytkownika

## Zabezpieczenia

Dostęp publiczny
/api/v1/parking-slots/state
/api/v1/auth/**

Wymagana rola ADMIN
/api/v1/users/{login}/toggleActive

Wszystkie inne wymagają AUTENTYKACJI (AUTH – Bearer Token)