**Paweł Warcaba**

**Grzegorz Żak**

**III rok, Informatyka**

**Aplikacji do tworzenia grup dyskusyjnych**

# Opis:

Naszym wspólnym pomysłem jest stworzenie „Aplikacji do tworzenia grup dyskusyjnych”, w której poza standardowymi funkcjonalnościami takimi jak logowanie, rejestracja i podział na „Administratora”, „Moderatora” i „Użytkownika”, istnieją grupy (tzw. pokoje). Każda grupa wymaga prośby o dołączenie lub zgłoszenia. W przypadku zgłoszenia akceptacji ze strony „Moderatora”. Każdy „Użytkownik” ma prawa do tworzenia grup. W każdej z grup będzie możliwość przeprowadzania rozmów na dany temat, a także możliwość wrzucania linków czy linków do obrazków. Użytkownicy mogą pisać wiadomości tylko w obrębie grupy do której należą. Główną zasadą naszego serwisu jest to, że z wielką łatwością każdy „Użytkownik ma możliwość w prosty sposób utworzyć grupy i zapraszać do nich użytkowników serwisu. Twórca grupy „Moderator” może usuwać nieodpowiednie osoby z grupy. Administrator serwisu ma wgląd we wszystkie grupy i może permanentnie ją zamknąć (np. jeśli narusza podstawowe zasady etyki). Każda osoba w naszym serwisie będzie miała swój profil wraz z możliwościami ustawienia swoich danych osobowych tzw. profil użytkownika.

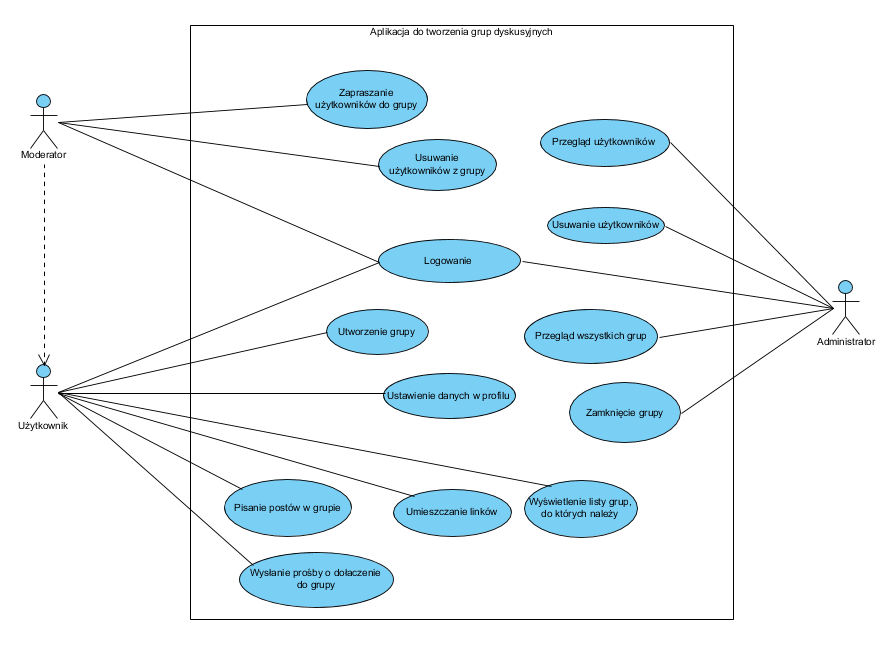
# Technologie:

Strona internetowa będzie napisana w technologii React.js (Front-end), domyślnie Node.js (Back-end). Oczywiście serwis będzie zawierał bazę danych PostgreSQL.

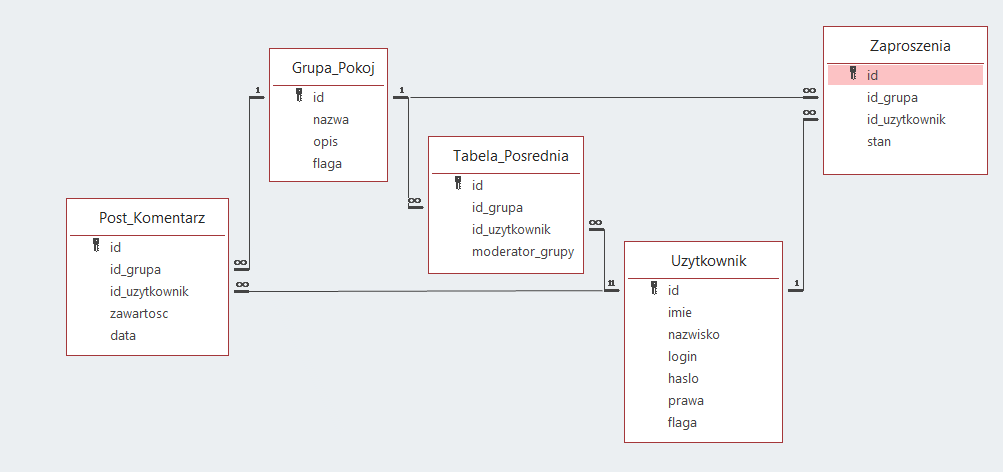
# Funkcjonalności:

* **„Administrator” witryny:**
  + Ustawienie danych w profilu
  + Wyświetlenie danych w profilu
  + Przegląd wszystkich grup
  + Zamknięcie grup (Nie ma możliwości jej wyświetlenia)
  + Przegląd użytkowników
  + Blokada użytkowników (Nie ma możliwości zalogowania się)
  + Wyświetlanie wszystkich grup (Automatycznie zostaje „Moderatorem” nowo utworzonej grupy)
  + Tworzenie (pisanie) „Postów” we wszystkich grupach
  + Wyświetlanie „Postów” ze wszystkich grup
* **„Moderator” Grupy/Pokoju:**
  + Ustawienie danych w profilu
  + Wyświetlenie danych w profilu
  + Dbałość o zachowanie porządku w grupie
  + „Akceptacja” lub „Blokowanie” prośby o dołączenie użytkowników do jego grupy, w której jest moderatorem
  + Blokowanie użytkowników z własnej grupy
  + Wyświetlenie listy rozwijanej (Dropdown) z wyborem nazwy grupy, w której jest moderatorem, a następnie tabeli użytkowników i moderatorów, którzy należą do jego grup.
  + Wyświetlanie grup do których należy lub jest moderatorem
  + Tworzenie (pisanie) „Postów” w grupie do której należy lub jest moderatorem
  + Wyświetlanie „Postów” z danej grupy do której należy lub jest moderatorem
* **Zwykły „Użytkownik”:**
  + Ustawienie danych w profilu
  + Wyświetlenie danych w profilu
  + Tworzenie grup (automatycznie nadaje prawo „Moderatora” w tej grupie)
  + Wysłanie prośby o dołączenie do danej grupy (z listy)
  + Wyświetlanie grup do których należy
  + Tworzenie (pisanie) „Postów” w grupie do której należy
  + Wyświetlanie „Postów” z danej grupy do której należy
  + Możliwość wrzucania obrazków i gifów – jako nieobowiązkowy załącznik wiadomości

# Diagram przypadków użycia:



# Baza danych:



* Żadne pole w naszej bazie danych nie może przyjąć wartości NULL.
* Pole „login” posiada wartość unikalną dla każdego użytkownika.
* Pole „prawa” przechowuje prawa dla użytkownika Administrator/Użytkownik. Pierwszy użytkownik, który utworzy konto otrzymuje prawa Administratora. Każdego kolejne konto jest zwykłym „Użytkownikiem”.
* Pierwszy Użytkownik tworzący grupę otrzymuje flagę „True” w polu „moderator\_grupy”.
* Pole „stan” przechowuje stan dla użytkownika Oczekujące/Zaakceptowane/Odrzucone.
* Pole data w tabeli „Post\_Komentarz” przechowuje datę
* Flaga w tabelach „Uzytkownik” i „Grupa\_Pokoj” dla funkcjonalności blokowania

# Instalacja i uruchomienie aplikacji:

1. Instalujemy Node.js

2. Instalujemy i uruchamiamy bazę danych PostgreSQL

a) Ustawienia dotyczące bazy danych:

**module.exports = {**

**pgUser: "postgres",**

**pgHost: process.env.PG\_HOST,**

**pgDatabase: "project",**

**pgPassword: "admin",**

**pgPort: process.env.PG\_PORT**

**};**

b) Plik konfiguracyjny powinien znajdować się w: C:\…\PPS\my-app\backend\keys.js

**Jeśli podany plik nie istnieje należy go utworzyć o nazwie „keys.js” (Plik nie znajduje się w repozytorium ponieważ takie pliki (m. in. z kluczami np. API) są wrażliwe i nie są upublicznianie w repozytorium githuba.com)**

4. Tworzymy baza danych o nazwie **project** (użytkownik i hasło musi być takie same)

5. Otwieramy dwie konsole Node.js dla backend i frontend

a) W pierwszej konsoli przechodzimy do ścieżki z backendem poleceniem:

**cd C:\...\PPS\my-app\backend**

Instalujemy dependencje:

**npm install lub npm i**

Następnie włączamy serwer **Node.js** poleceniem:

**nodemon**

b) W drugiej konsoli przechodzimy do ścieżki z frontendem poleceniem:

**cd C:\...\PPS\my-app\frontend**

Instalujemy dependencje:

**npm install lub npm i**

Następnie włączamy widok **React.js** poleceniem:

**npm start**

6. Automatycznie w przeglądarce powinna wystartować nasza witryna

7. Projekt znajduje się na github.com, link do repozytorium:

<https://github.com/pawel54321/PPS>

Spis treści

[Opis: 1](#_Toc9534577)

[Technologie: 1](#_Toc9534578)

[Funkcjonalności: 1](#_Toc9534579)

[Diagram przypadków użycia: 3](#_Toc9534580)

[Baza danych: 3](#_Toc9534581)

[Instalacja i uruchomienie aplikacji: 4](#_Toc9534582)