

Opis systemu

Celem systemu będzie informowanie użytkowników o różnego rodzaju wydarzeniach dziejących się w mieście, które wybierze użytkownik. Dodatkowo, po zalogowaniu, użytkownik będzie mógł wyrazić chęć oraz uczestnictwo w poszczególnym wydarzeniu. Użytkownicy powinni móc tworzyć swoje wydarzenia oraz zapraszać na nie innych użytkowników zarejestrowanych w systemie. Aplikacja powinna pozwalać również na zestawienie kalendarzowe wydarzeń oraz mapę ze znacznikami.

Lista funkcji

Użytkownik (niezalogowany):

- Rejestracja
- Logowanie
- Przeglądanie wydarzeń
 - Lista wydarzeń
 - Mapa ze znacznikami wydarzeń
 - Filtrowanie wydarzeń
- powiadomienie o nowym wydarzeniu, dla wybranego miasta

Użytkownik (zalogowany):

To samo, co użytkownik niezalogowany oraz:

- Wylogowanie
- Edycja własnego profilu:
 - imię
 - nazwisko
 - podstawowe miasto
 - preferowane wydarzenia
 - modyfikacja hasła
 - usunięcie konta
- Tworzenie, modyfikacja wydarzeń
 - Nazwa,
 - Krótki opis wydarzenia
 - Miasto
 - Data, godzina, czas trwania
 - Predefiniowany rodzaj wydarzenia
- Odwoływanie własnych wydarzeń
- Możliwość wyrażenia chęci uczestniczenia w wydarzeniu

Administrator

To samo, co użytkownik zalogowany oraz:

- Możliwość odwołania dowolnego wydarzenia w systemie
- Możliwość anonimizacji(usunięcia) użytkownika
- Modyfikacja danych użytkownika
- Modyfikacja danych wydarzeń

Diagram hierarchii funkcji

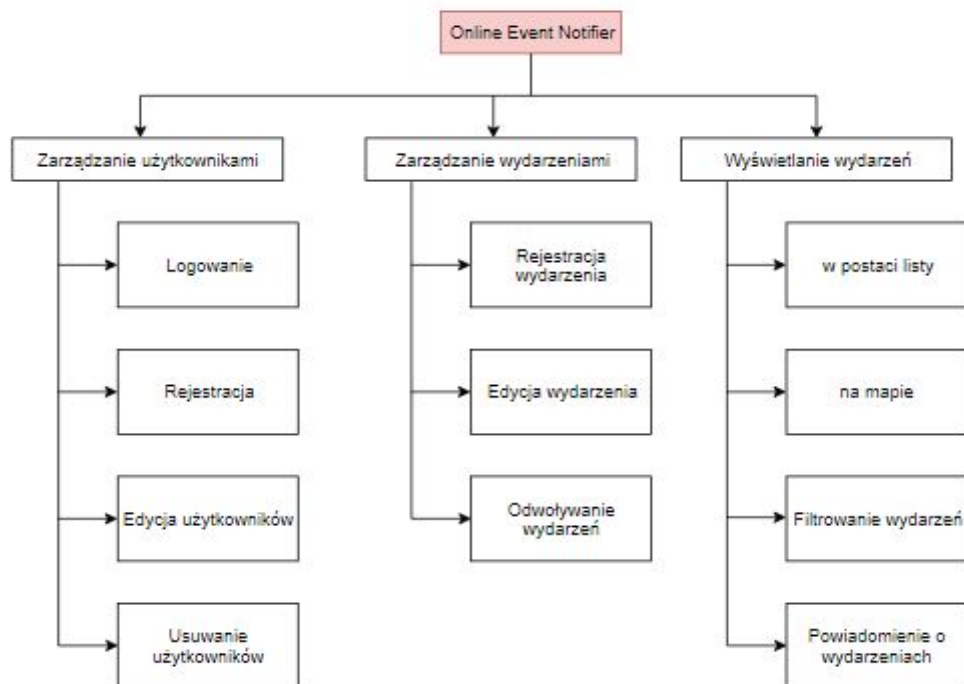


Diagram przypadków użycia

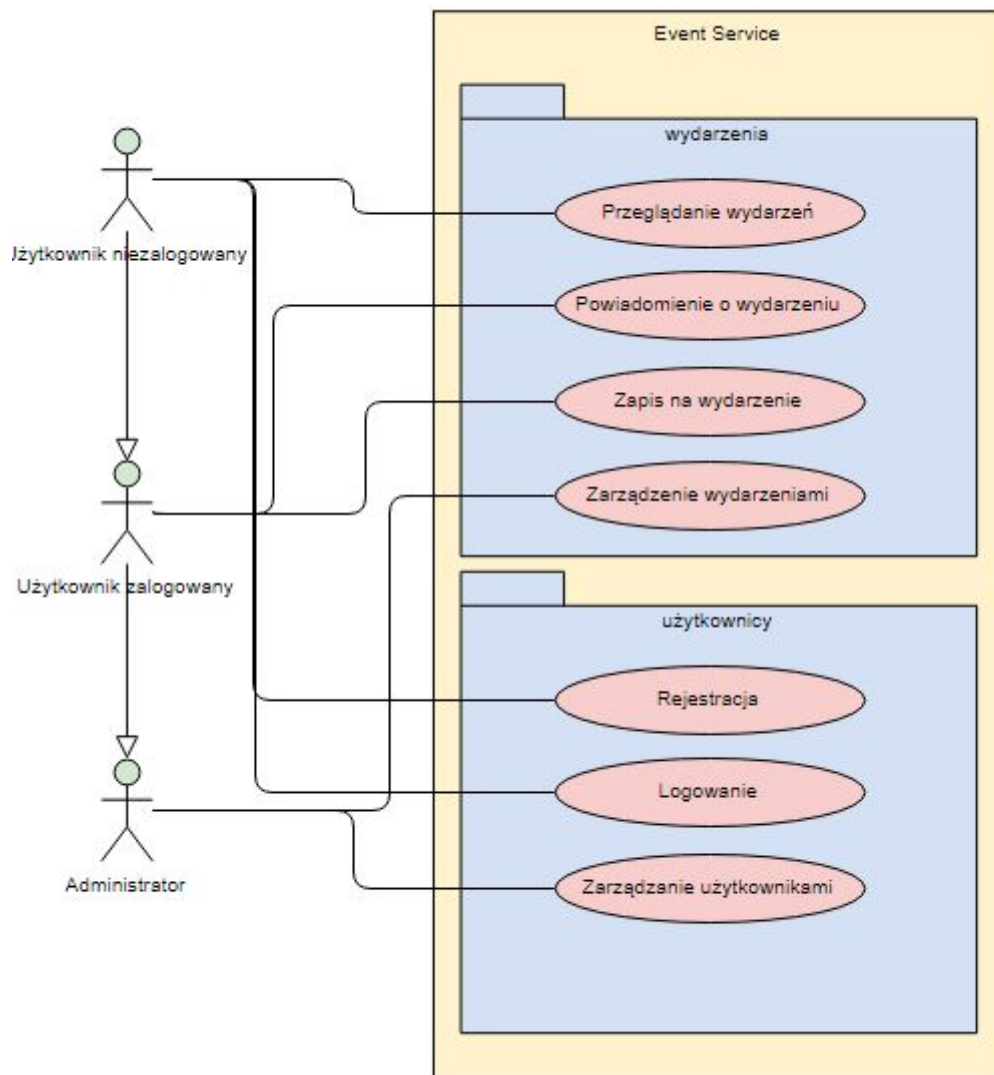
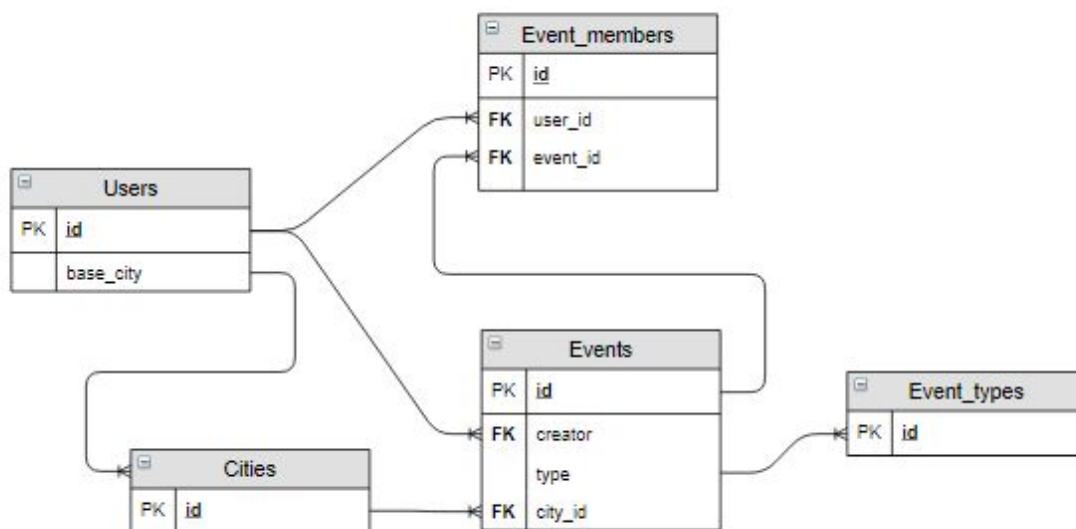


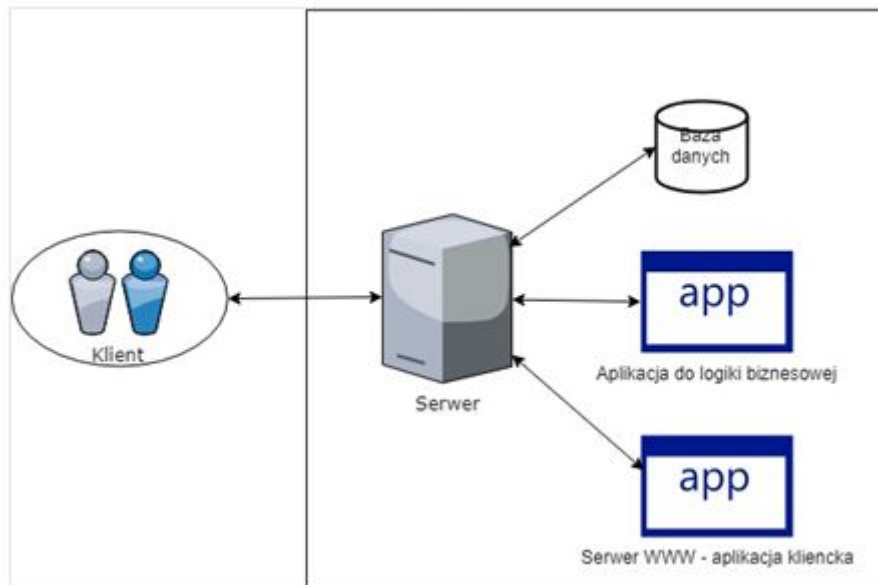
Diagram ERD



Users->id

Model architektury systemu

W projekcie zostanie wykorzystany model komunikacji typu klient-serwer, a co za tym idzie - powstanie struktura rozproszona systemu, składająca się z połączenia dwóch osobnych aplikacji (frontend i backend) wraz ze wspierającą je bazą danych, gdzie użytkownik za pomocą aplikacji front-endowej będzie komunikować się z serwerem, a dokładniej aplikacją back-endową.



Rysunek: Architektura typu klient-serwer, źródło: opracowanie własne

Dodatkowo aplikacja będzie korzystać ze wzorca MVC, który ułatwi tworzenie aplikacji internetowej oraz w sposób logiczny dzieli jej elementy na tzw. Model-widok-kontroler.

Do komunikacji pomiędzy aplikacją front-endową i back-endową będzie wykorzystywane tzw. REST API, które wystawi aplikacja back-endowa.

Front-endowe technologie spełniające powyższe wymagania to np.: React, Angular, Vue
Back-endowe technologie spełniające powyższe wymagania to np.: PHP(Laravel), Java(Spring) oraz .NET.

Do projektu zostały wybrane React oraz PHP(Laravel), gdyż każdy uczestnik projektu posiada już pewną wiedzę na temat tych frameworków.