PROJEKT I IMPLEMENTACJA SYSTEMU POWIADOMIEŃ ONLINE O IMPREZACH REALIZOWANYCH W MIEŚCIE

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.

Rodzaje użytkowników w systemie

W systemie występować będą następujący użytkownicy:
Użytkownik niezalogowany - jest to użytkownik domyślny systemu, posiadający najmniejszą ilość uprawnień. Głównym celem tego użytkownika będzie możliwość sprawdzenia wydarzeń, jakie są w wybranym przez niego mieście.
Użytkownik zalogowany - jest rozszerzeniem użytkownika niezalogowanego, gdyż posiada dodatkowe uprawnienia, np: możliwość tworzenia wydarzeń, możliwość wyrażenia chęci uczestnictwa w wydarzeniu.
Administrator - jest rozszerzeniem użytkownika zalogowanego i jednocześnie użytkownikiem z największymi uprawnieniami - względem użytkownika zalogowanego posiada dodatkowe uprawnienia, np: możliwość modyfikacji, usunięcia dowolnego wydarzenia lub użytkownika.

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:

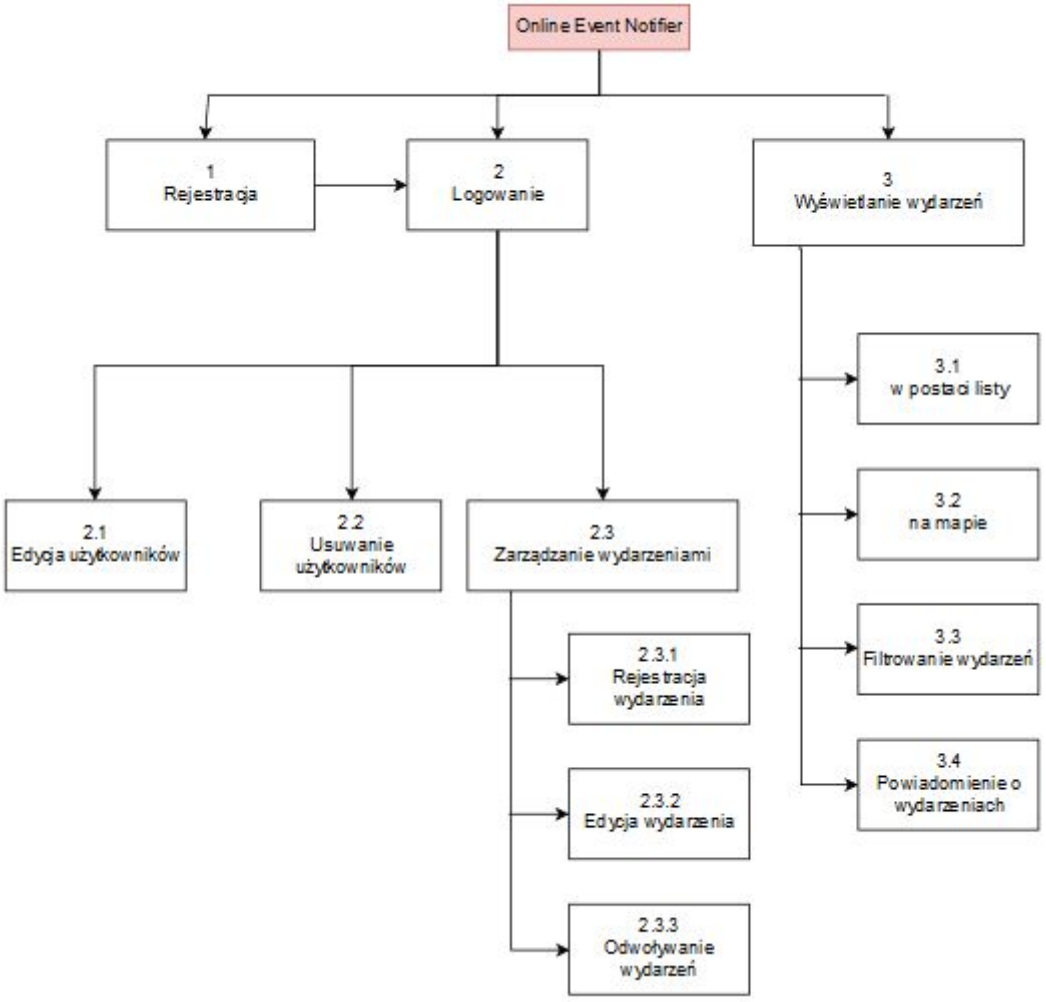
        - imię
        - nazwisko
        - podstawowe miasto
        - preferowane wydarzenia
        - modyfikacja hasła
        - usunięcie konta
    - Tworzenie, modyfikacja wydarzeń
        - Nazwa,
        - Krótki opis wydarzenia
        - Miasto
        - Data, godzina, czas trwania
        - Predefiniowany 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



rys.1 Diagram hierarchii funkcji

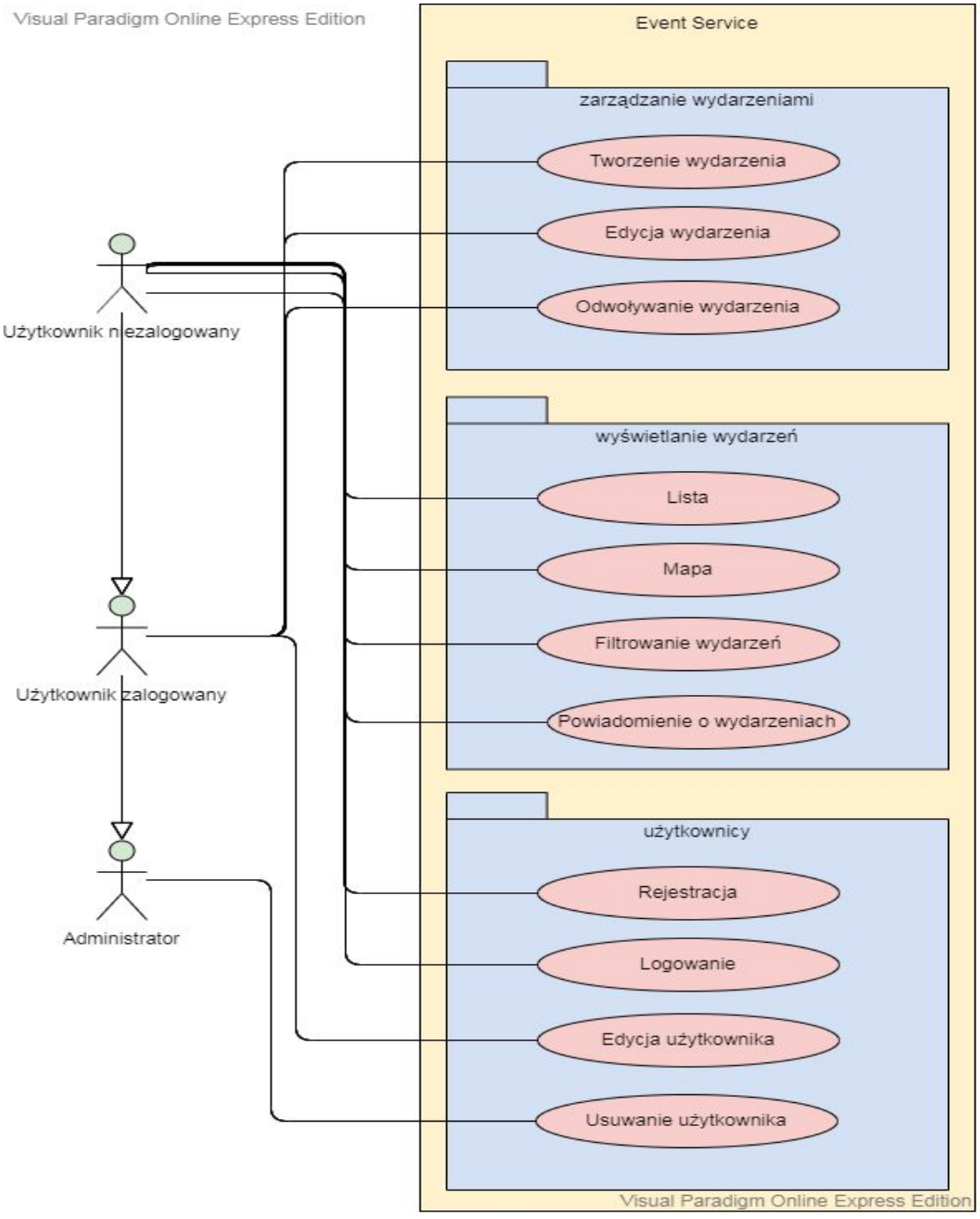
System posiada trzy rodzaje funkcji { rys.1}:

-

lp.	Funkcje	Opis
1	Rejestracja	Rejestracja nowego użytkownika
2	Logowanie	Logowanie już zarejestrowanego użytkownika
3	Wyświetlanie wydarzeń	Funkcja wyświetlania informacji o wydarzeniach
2.1	Edycja użytkowników	Edycja informacji zawartych w profilu użytkownika
2.2	Usuwanie użytkowników	Usuwanie konta użytkownika
2.3	Zarządzanie wydarzeniami	Zarządzanie wydarzeniami w systemie

2.3.1	Rejestracja wydarzenia	Rejestracja nowego wydarzenia w systemie
2.3.2	Edycja wydarzenia	Edycja wydarzenia w systemie
2.3.3	Odwoływanie wydarzeń	Odwoływanie wydarzeń w systemie
3.1	W postaci listy	Wyświetlanie wydarzeń w postaci listy
3.2	na mapie	Wyświetlanie wydarzeń jako znaczniki na mapie
3.3	Filtrowanie wydarzeń	Filtrowanie wydarzeń za pomocą wcześniej zdefiniowanych filtrów
3.4	Powiadomień o wydarzeniach	Powiadomienie użytkowników zalogowanych oraz niezalogowanych o nowych wydarzeniach

Diagram przypadków użycia



rys.2 Diagram przypadków użycia

Dostępni aktorzy

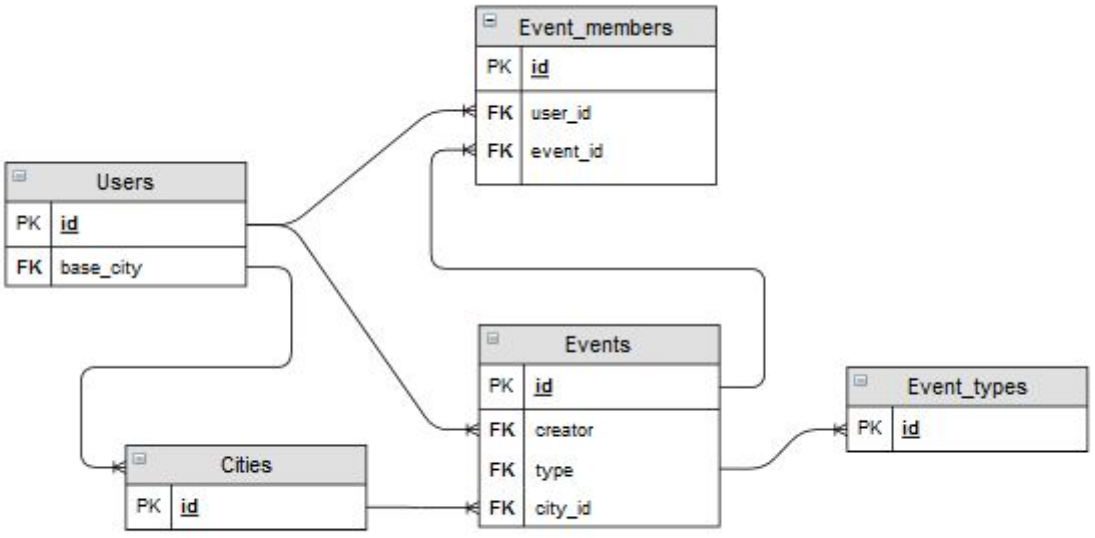
Użytkownik niezalogowany - ma dostęp do podstawowych funkcjonalności serwisu, między innymi: założenie nowego konta, przeglądanie dostępnych wydarzeń w formie listy

lub mapy, filtrowanie wyników uwzględniając lokalizację oraz typ wydarzenia. Dodatkowo niezalogowany użytkownik może zapisać się na nieograniczoną liczbę wydarzeń.

Użytkownik zalogowany - ma dostęp do wszystkich funkcji, które posiada użytkownik niezalogowany, jednak dodatkowo może tworzyć nowe wydarzenia, edytować te, który już zostały przez niego utworzone, oraz odwoływać zdarzenia, które się nie odbędą.

Administrator - posiada pełne uprawnienia do wszystkich funkcjonalności serwisu { rys.2}

Diagram ERD



rys.3 Diagram ERD

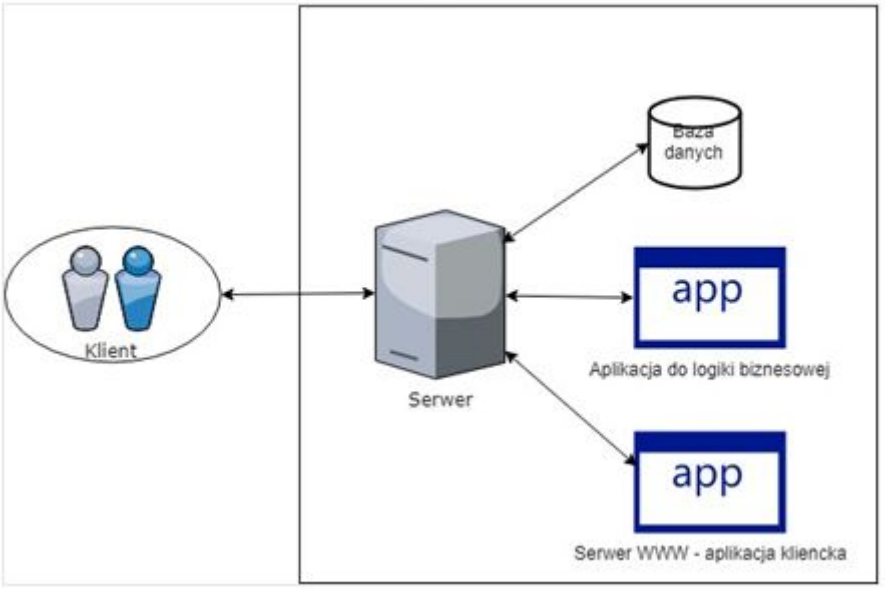
System będzie posiadać następujące tabele {rys 3.}:

- Users - tabela posiadająca informacje o zalogowanych użytkownikach systemu,
- Events - tabela posiadająca informacje o wszystkich wydarzeniach oraz informacje o autorze wydarzenia,
- Event_members - tabela posiadająca informacje o użytkownikach zainteresowanych danym wydarzeniem,
- Cities - tabela posiadająca informacje o potencjalnych miastach dla wydarzeń,
- Event_types - tabela posiada informacje o typach wydarzeń.

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.



rys.4 Diagram architektury systemu.

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.

Etap II

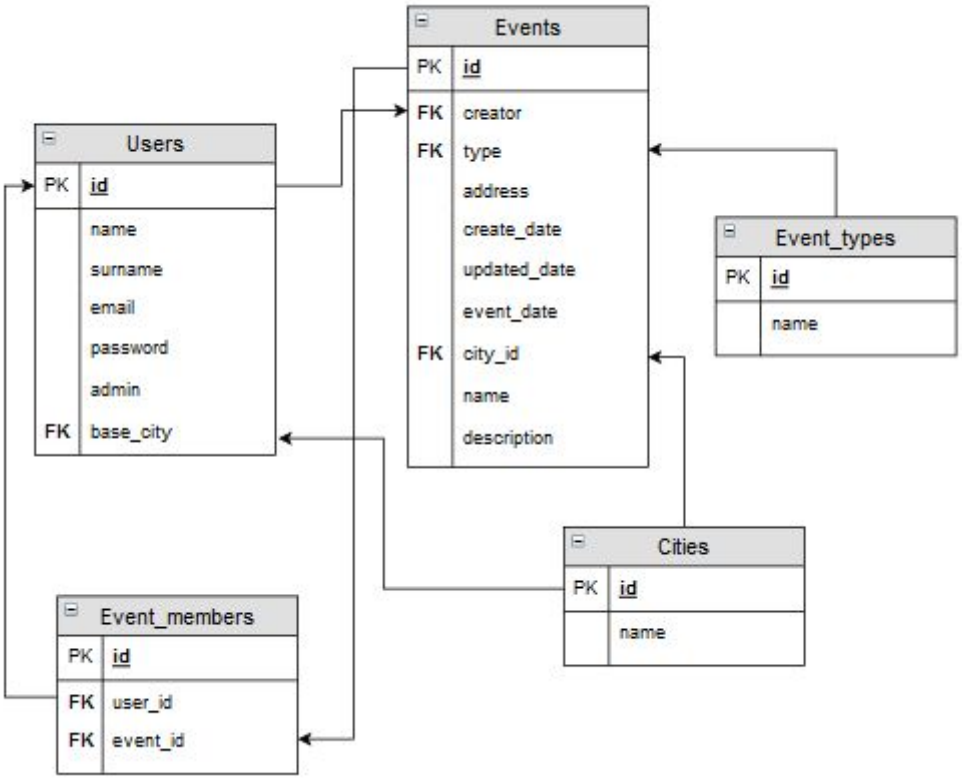
Projekt architektury systemu

Projekt bazy danych

W poniższym podrozdziale zawarty został projekt bazy danych, czyli: opis bazy danych, diagramy tabel i relacji, opisy pól i ich atrybutów, opisy relacji oraz opis mechanizmów bazy danych.

Opis bazy danych
Baza danych zostanie wykonana w systemie zarządzania bazami danych MySQL.

Diagram tabel



Baza danych będzie składać się z następujących encji:

Encja Users - będzie przechowywać informacje o użytkownikach w systemie.

Users			
Pole	Typ	Atrybuty	Opis
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
name	VARVHAR(255)	NOT_NULL	Imię użytkownika
surname	VARCHAR(255)	NOT_NULL	Nazwisko użytkownika
email	VARCHAR(255)	NOT_NULL	Email użytkownika
password	NCHAR(60)	NOT_NULL	Zakodowane hasło użytkownika
admin	TINYINT(1)	NOT_NULL	Flaga określająca, czy użytkownik posiada uprawnienia administratora
base_city	INT	NOT_NULL,	Domyślne miasto, które będzie wyświetlane użytkownikowi, zanim ten zmieni kryteria wyświetlania wydarzeń.

Encja Events zawiera informacje o wydarzeniach.

Events			
Pole	Typ	Atrybuty	Opis
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
creator	INT	NOT_NULL, FORIEGN_KEY	Id użytkownika, który stworzył wydarzenie
type	INT	NOT_NULL, FOREIGN_KEY	Id typu wydarzenia
address	VARCHAR(1024)	NOT_NULL	Adres wydarzenia
create_date	DATETIME	NOT_NULL	Data utworzenia wydarzenia

updated_date	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o wydarzeniu
event_date	DATETIME	NOT_NULL	Data, kiedy odbędzie się wydarzenie
city_id	INT	NOT_NULL, FOREIGN_KEY	Id miasta, w którym ma się odbyć wydarzenie
name	VARCHAR(255)	NOT_NULL	Nazwa wydarzenia
description	VARCHAR(512)	NOT_NULL	Opis wydarzenia

Encja Cities zawiera nazwę miastach.

Cities			
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
name	VARVHAR(255)	NOT_NULL	Nazwa miasta

Encja Event_members zawiera typ wydarzenia

Event_members			
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
user_id	INT	NOT_NULL	id_użytkownika
event_id	INT	NOT_NULL	id_wydarzenia

Encja Event_types zawiera typ wydarzenia

Event_types			
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
name	VARVHAR(255)	NOT_NULL	Typ wydarzenia

Diagram relacji