

# 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:

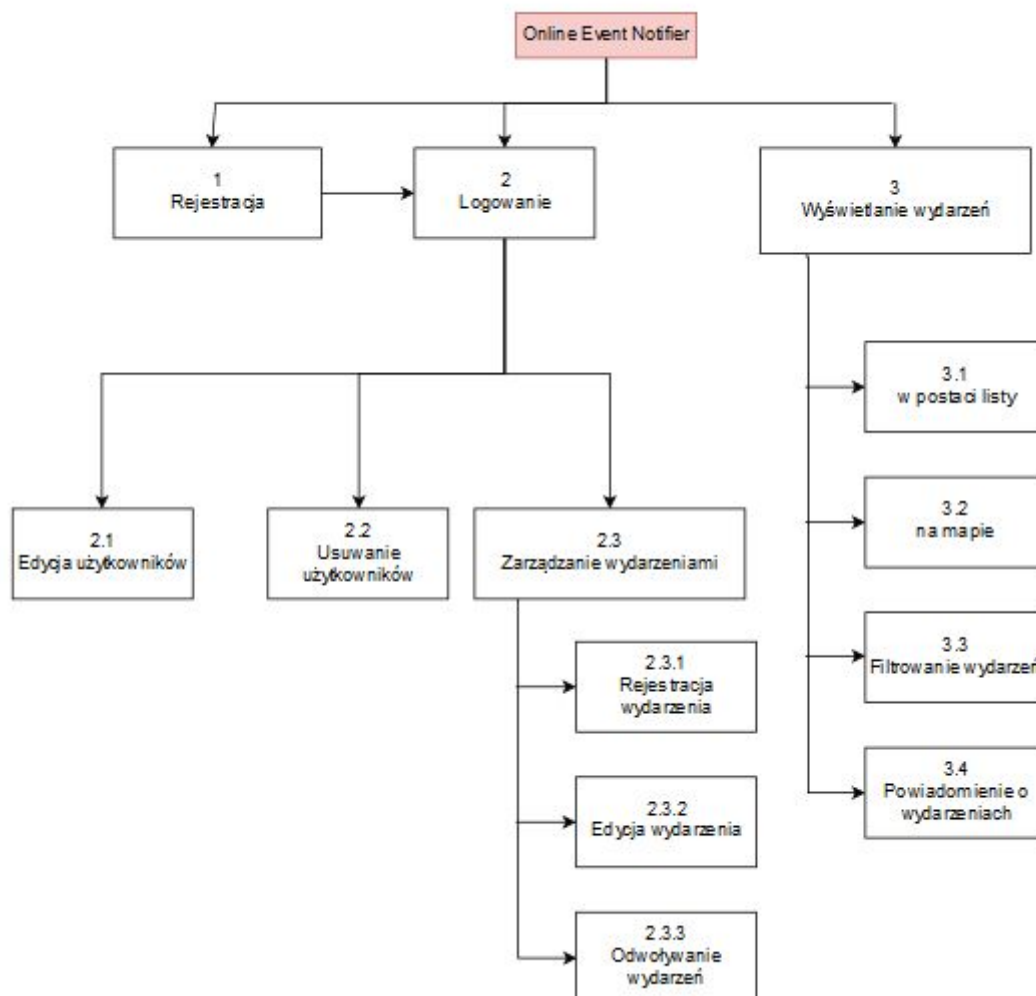
- imię
- nazwisko
- podstawowe miasto
- preferowane wydarzenia
- modyfikacja hasła
- usunięcie konta
- Tworzenie, modyfikacja wydarzeń
  - Nazwa,
  - Krótki opis wydarzenia
  - Miasto
  - Data, godzina, czas trwania
  - Predefiniowany rodzaj wydarzenia
- Odwoływanie własnych wydarzeń
- Możliwość wyrażenia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



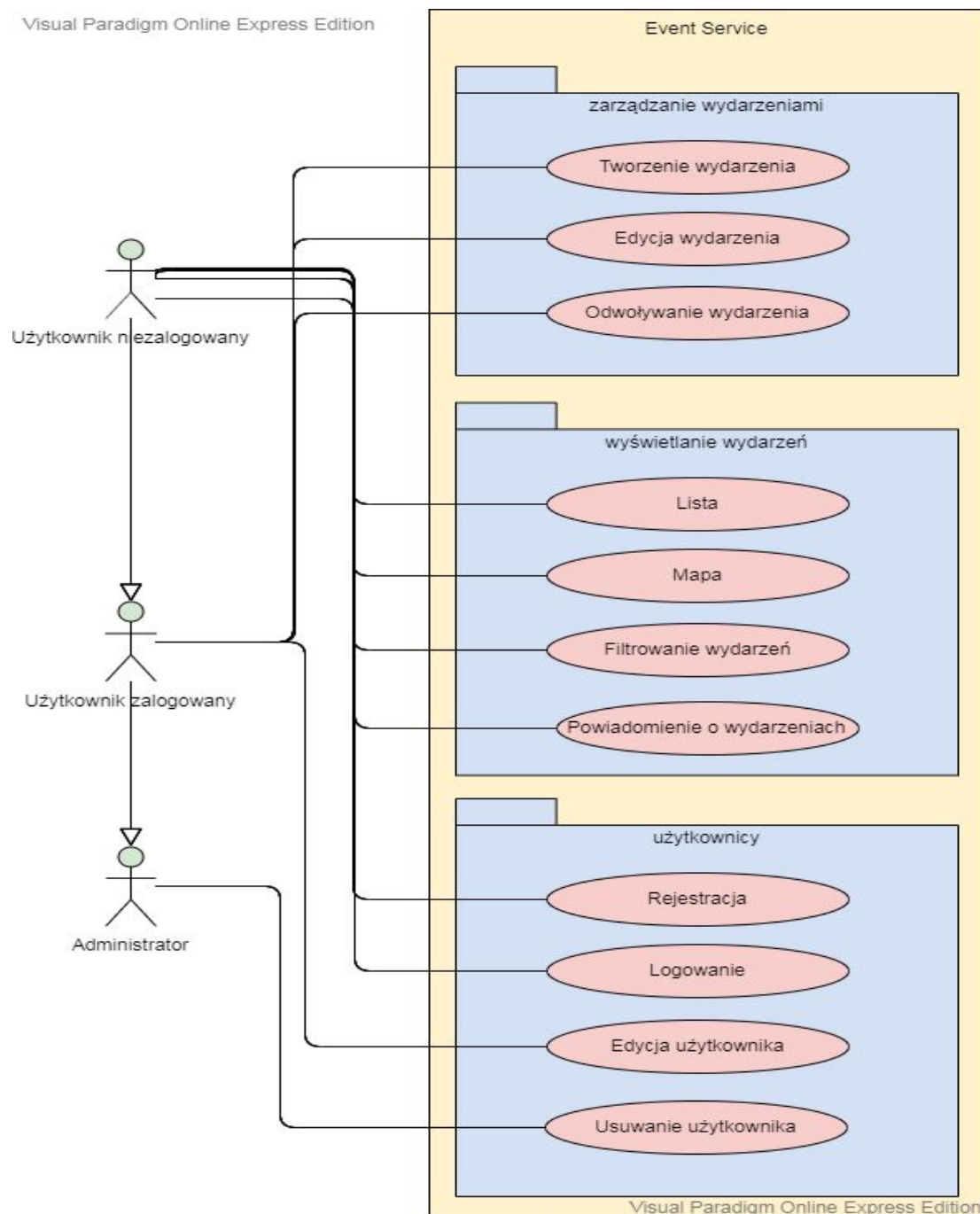
rys.1 Diagram hierarchii funkcji

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

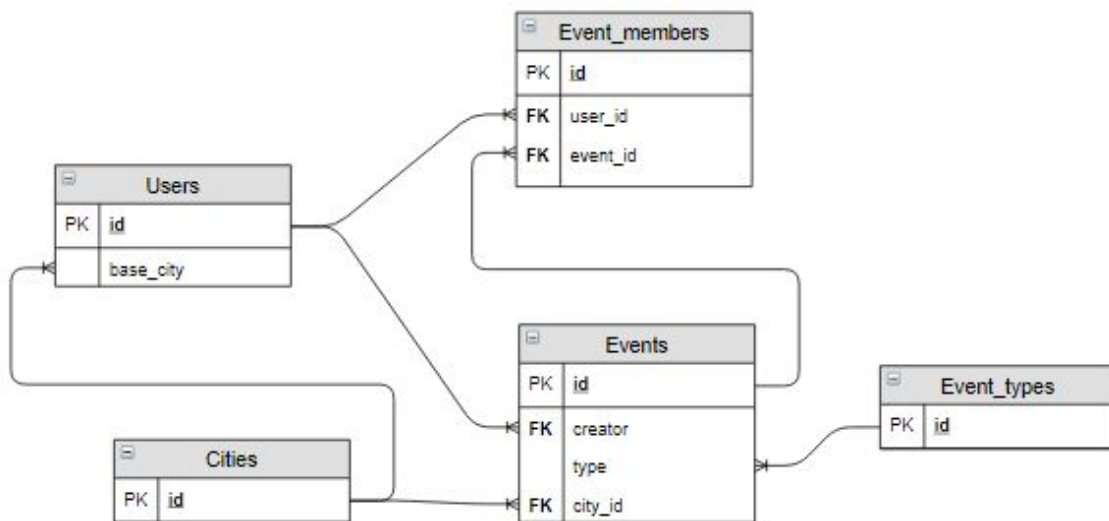
**Użytkownik niezalogowany** - ma dostęp do podstawowych funkcjonalności serwisu, między innymi: założenie nowego konta, przeglądanie dostępnych wydarzeń w formie listy

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

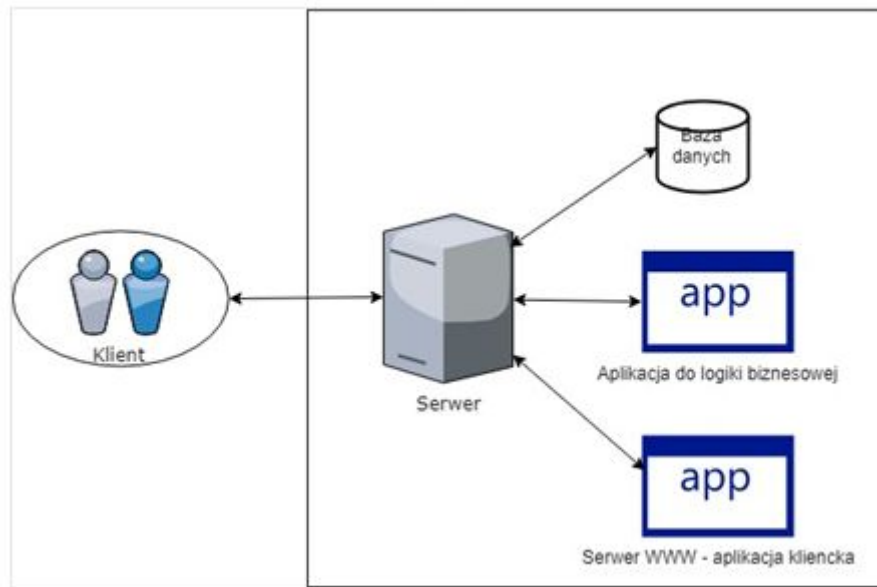
System będzie posiadać następujące tabele {rys 3.}:

- 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.

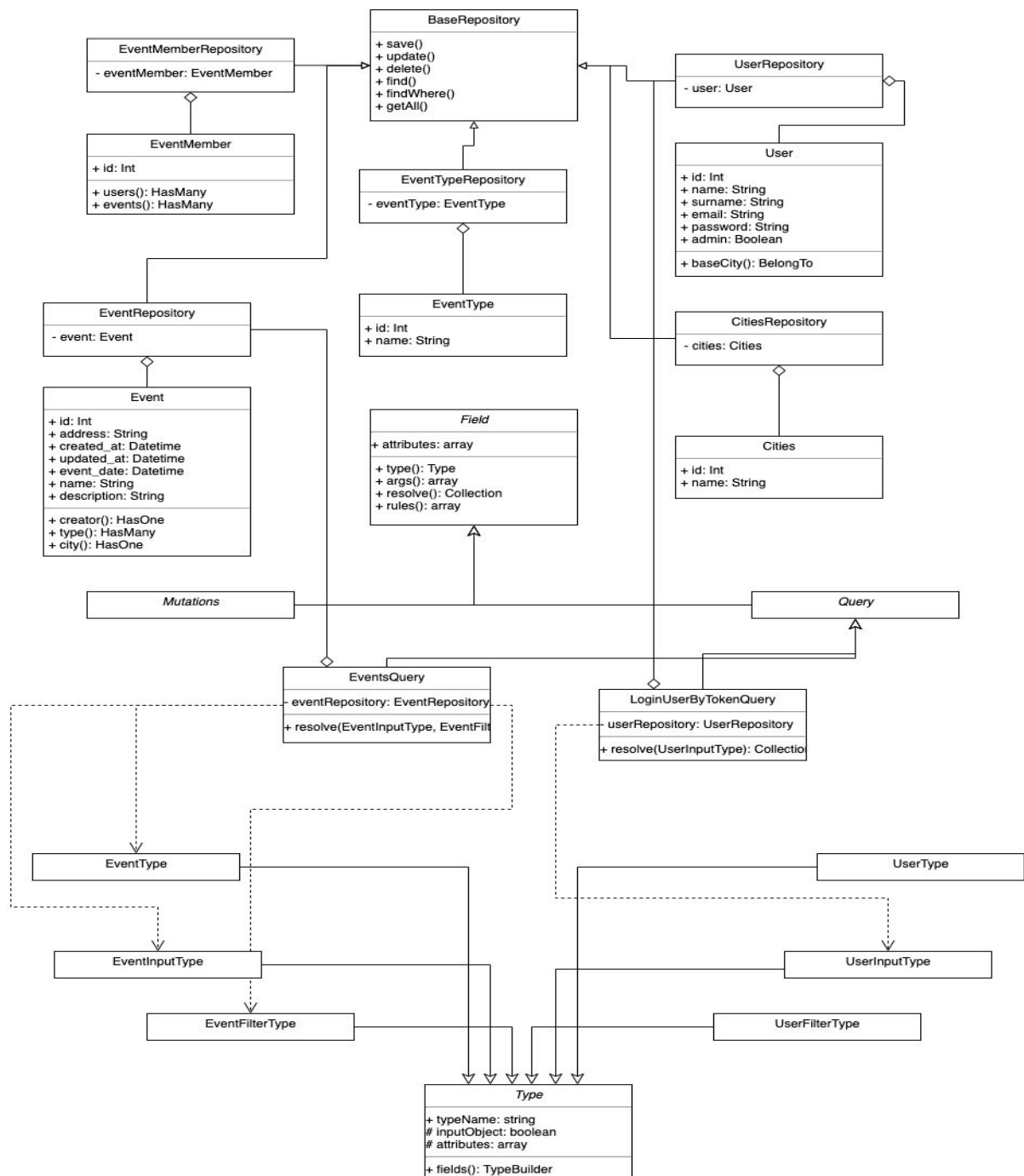
## Etap II

### Projekt architektury systemu

Diagram klas składa się z kilku głównych klas, które są dziedziczone w pozostałych klasach zapewniając całą potrzebną logikę

- BaseRepository - klasa abstrakcyjna, która zawiera wszystkie metody potrzebne do obsługi bazy danych. Implementacja klasy działa na dostarczonych modelach

- Field - klasa abstrakcyjna, która zapewnia obsługę api. Umożliwia rozwiązanie kontraktu poprzez metodę resolve. Definiuje również typy danych jakie może przyjąć poszczególny endpoint. Wszystkie te elementy są umieszczane już w konkretnej implementacji. Przykładowe klasy bazujące na Field: RegisterUserMutation, CreateEventMutation
- Type - klasa abstrakcyjna, pozwala dodawać poszczególne argumenty w klasach, które dziedziczą po niej dziedziczą, później będą wykorzystywane w klasie Field jako typy danych

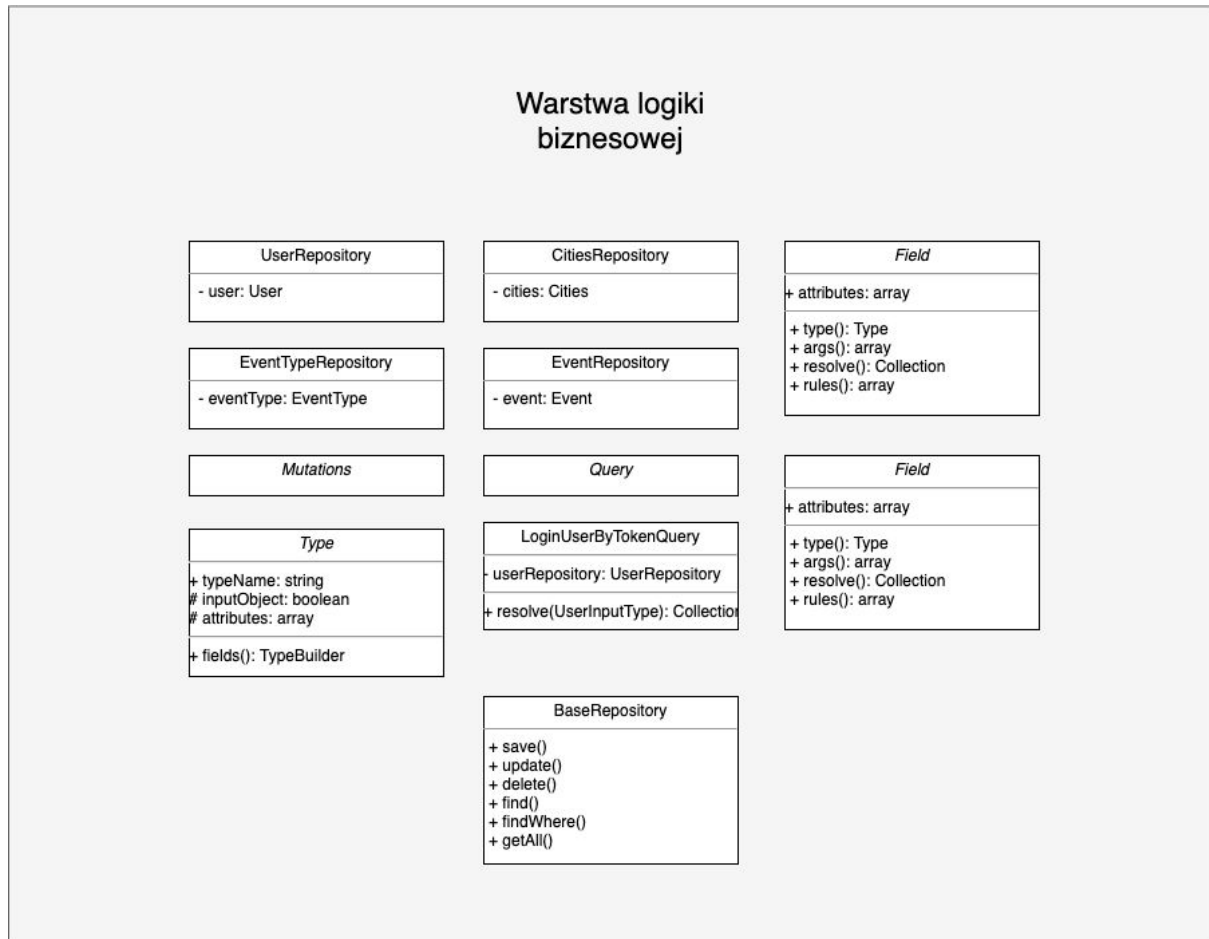


rys.5 Diagram klas



## Warstwa logiki biznesowej

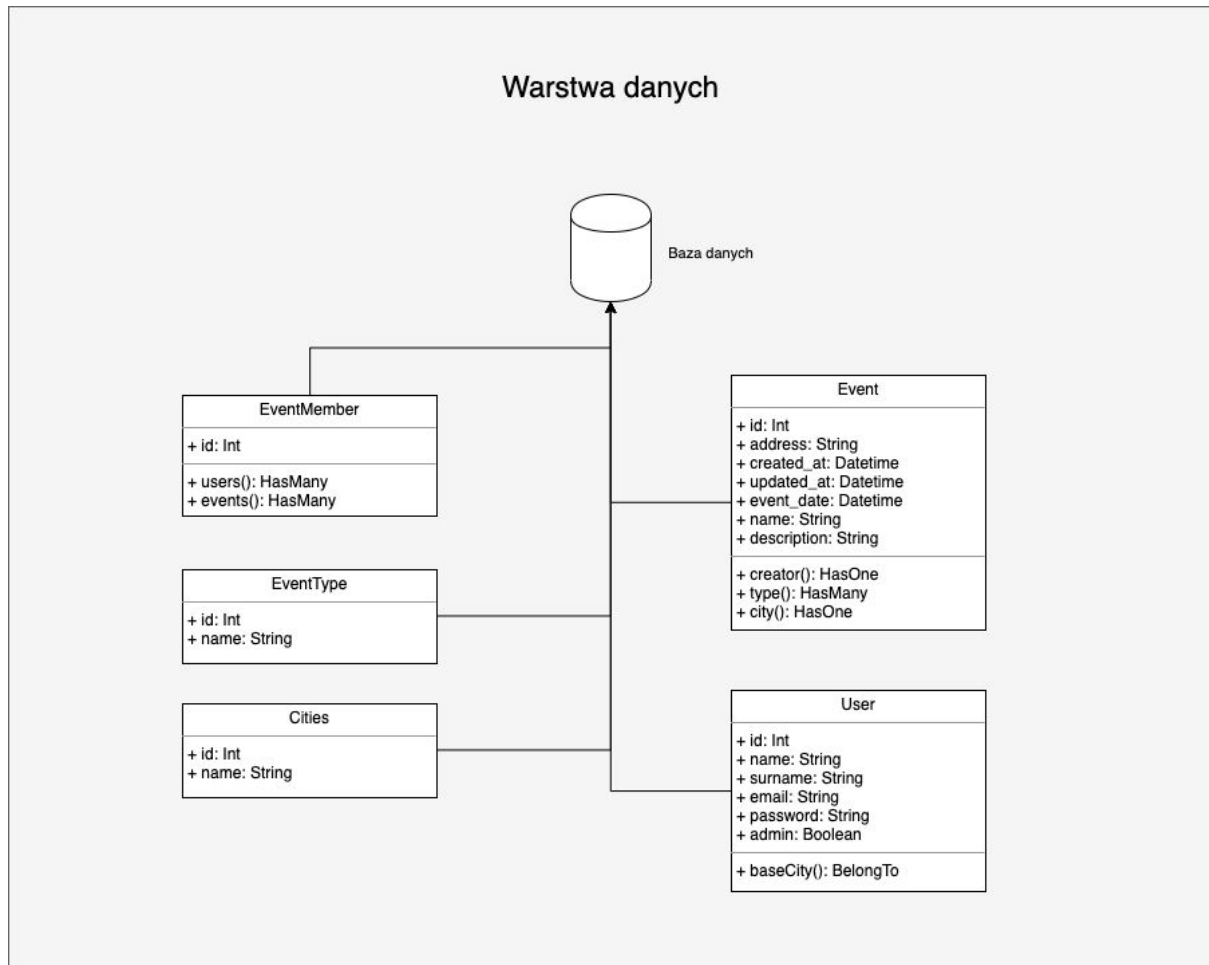
W warstwie logiki biznesowej, przechowywane są główne funkcje systemu, które zostały określone w wymaganiach funkcjonalnych, dzięki oddzieleniu tej warstwy, zapewniamy większe bezpieczeństwo oraz stabilność systemu. Warstwa ta jest pośrednią, pomiędzy warstwą prezentacji oraz warstwą danych



rys.6 Warstwa logiki biznesowej

## Warstwa danych

Warstwa ta zapewnia nam możliwość składowania, utrwalania, oraz pozyskiwania danych dla naszej aplikacji. Stanowi sposób reprezentacji danych w naszej aplikacji, zawiera klasy, właściwości, pola. może opisywać strukturę bazy danych



rys.7 Diagram warstwy danych

## Warstwa prezentacji

Jest to jedyna część, którą zobaczy końcowy użytkownik. W tej warstwie prezentowane są wcześniej przetworzone dane. Zapewnia również interakcję z użytkownikiem, oraz przekazywanie żądań do warstwy logiki. Nie wykonuje jakichkolwiek operacji na modelu danych. W naszym przypadku, za warstwę prezentacji będzie odpowiadał Framework ReactJs, który pozyska dane bezpośrednio z api w formacie json.



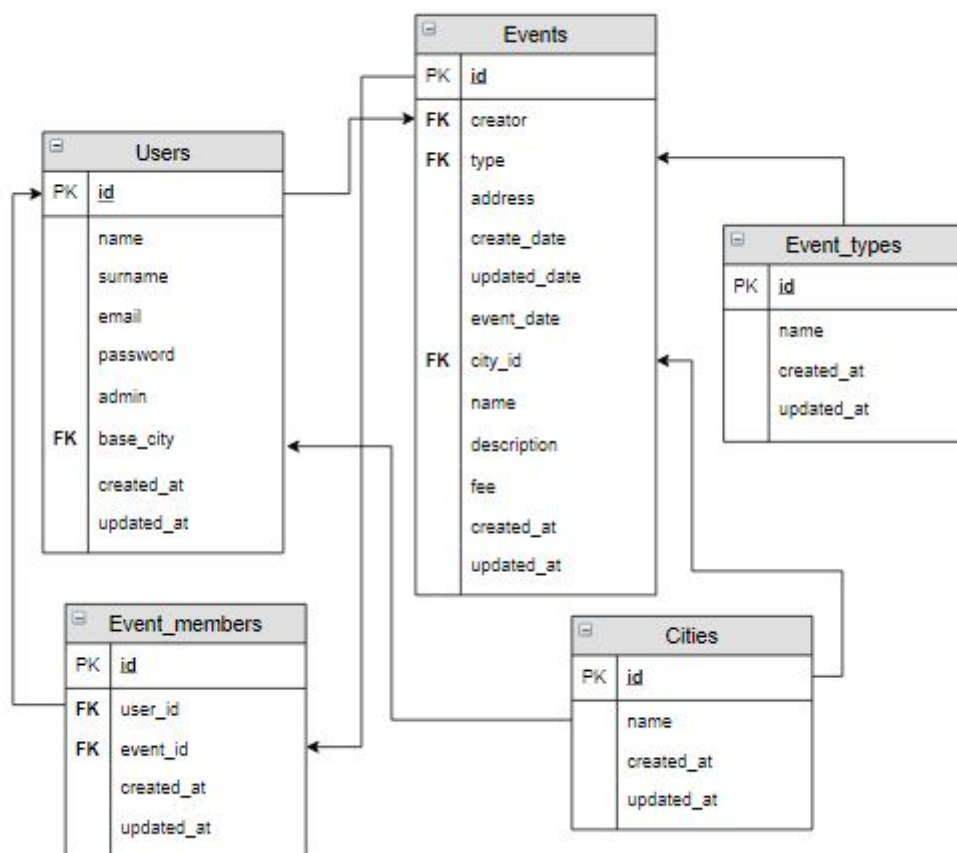
rys.8 Warstwa prezentacji

## 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.

Baza danych zostanie wykonana w systemie zarządzania bazami danych MySQL MariaDB.

## Diagram tabel



rys.9 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, FORIEGN_KEY	Domyślne miasto, które będzie wyświetlane użytkownikowi, zanim ten zmieni kryteria wyświetlania wydarzeń.
created_at	DATETIME	NOT_NULL	Data utworzenia użytkownika
updated_at	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o użytkowniku

Encja Events zawiera informacje o wydarzeniach.

<b>Events</b>			
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
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

fee	INT	NOT_NULL	Kwota pobierana na wejściu wydarzenia
created_at	DATETIME	NOT_NULL	Data utworzenia wydarzenia
updated_at	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o wydarzeniu

Encja Cities zawiera nazwę miastach.

Cities			
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
name	VARVHAR(255 )	NOT_NULL	Nazwa miasta
created_at	DATETIME	NOT_NULL	Data utworzenia miasta
updated_at	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o mieście

Encja Event\_members zawiera użytkowników zapisanych na wydarzenie

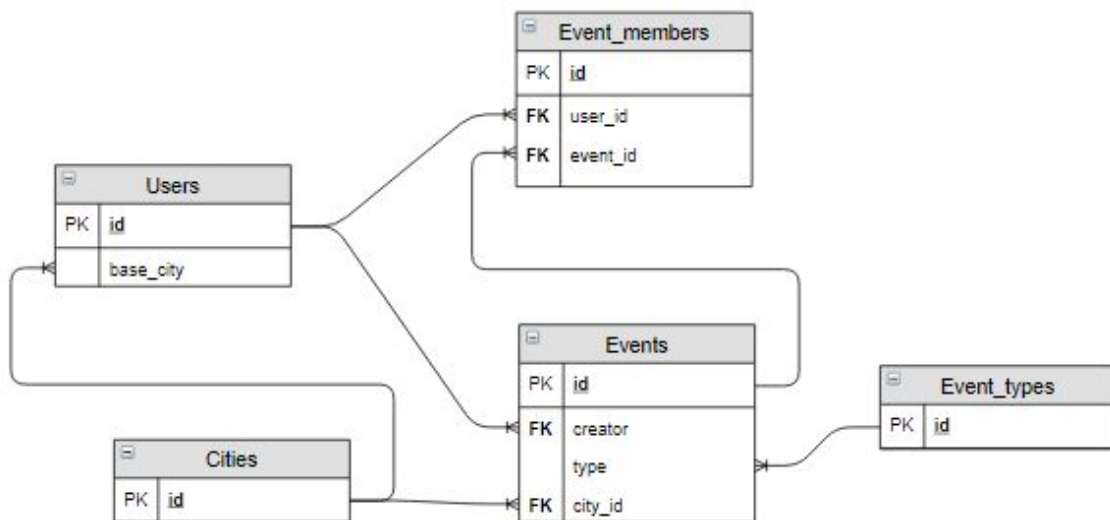
Event_members			
id	INT	AUTO_INCREMENT, PRIMARY_KEY, NOT_NULL	Unikalny klucz tabeli
user_id	INT	NOT_NULL, FORIEGN_KEY	id_użytkownika
event_id	INT	NOT_NULL, FORIEGN_KEY	id_wydarzenia
created_at	DATETIME	NOT_NULL	Data utworzenia wiązania
updated_at	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o wiązania

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

	)		
created_at	DATETIME	NOT_NULL	Data utworzenia typu
updated_at	DATETIME	NOT_NULL	Data ostatniej aktualizacji informacji o typie

## Diagram relacji



rys.10 Diagram relacji

Tabela	Pole	Tabela 2	Pole	Opis relacji
Users	id	Event_members	user_id	Jeden do wielu
Users	id	Events	creator	Jeden do wielu
Cities	id	Users	base_city	Jeden do wielu
Cities	id	Events	city_id	Jeden do wielu
Events	id	Event_members	event_id	Jeden do wielu
Event_types	id	Events	type	Jeden do wielu

W bazie nie będą występować żadne procedury składowane.

# Mechanizmy bazy danych

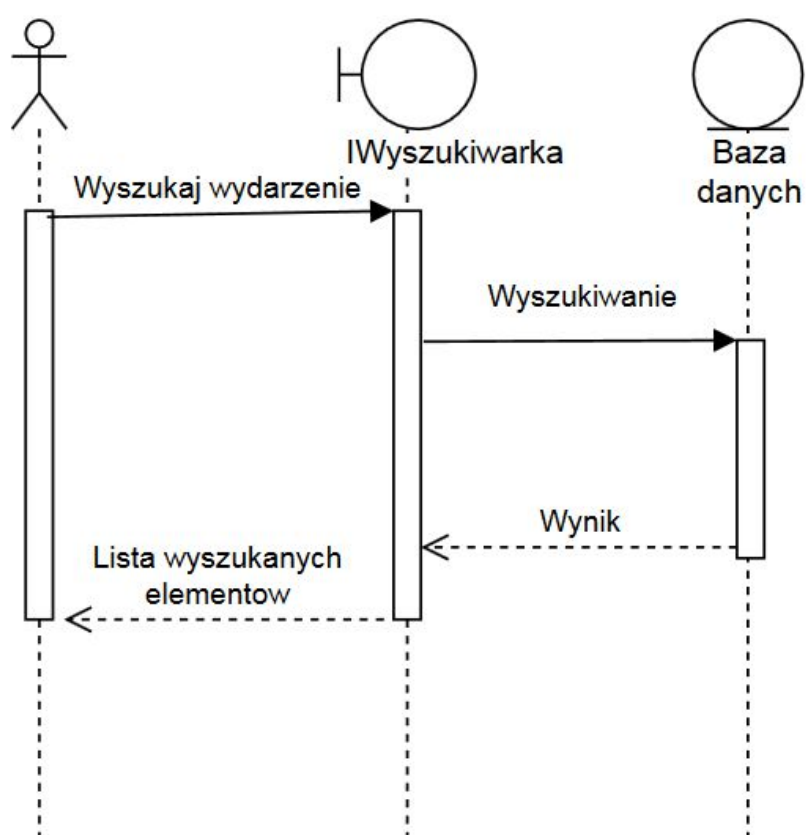
Do aplikacji dodane zostaną mechanizmy

- Narzędzi ORM
- Środki optymalizujące baze danych.
- Środki zapewniające integralność oraz bezpieczeństwo danych

## Projekt algorytmów

### Diagram sekwencji

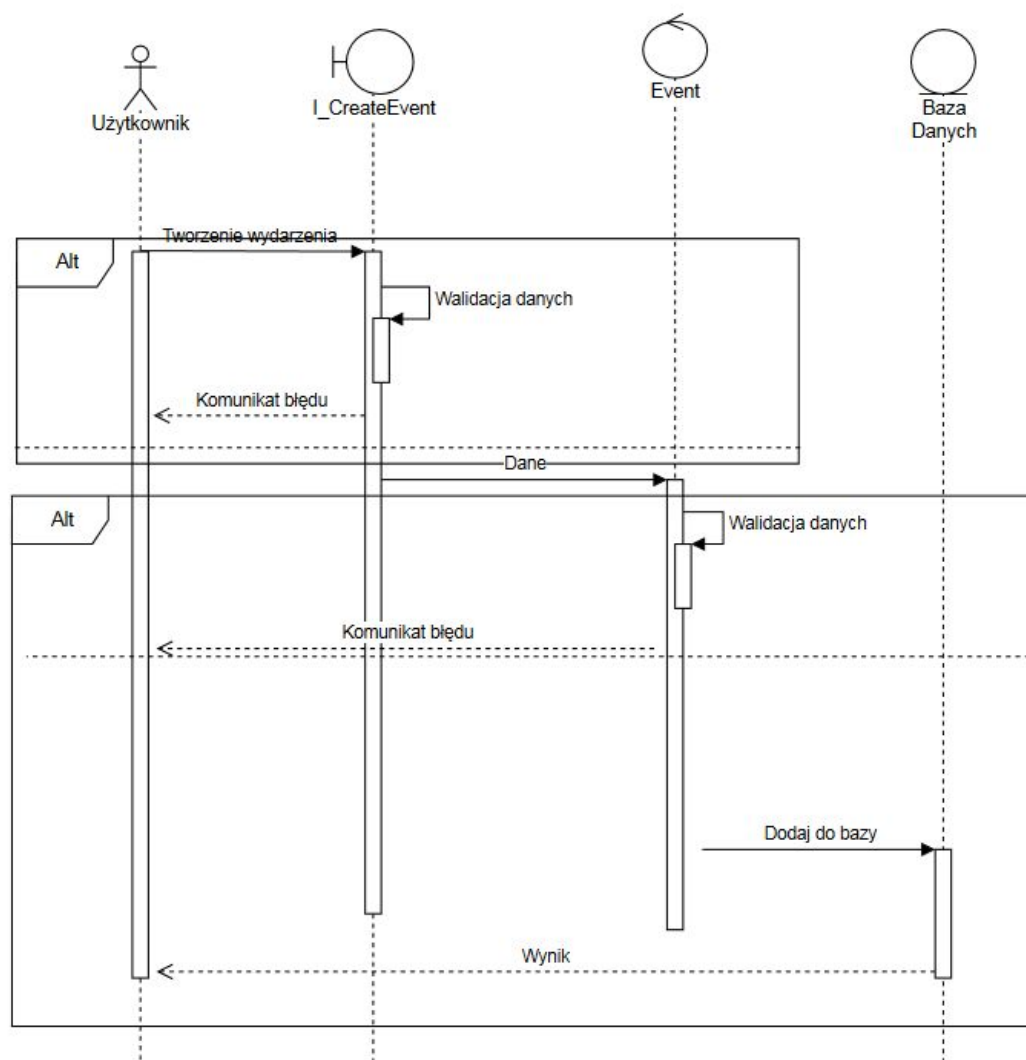
Diagram Wyszukiwania



rys.11 Diagram sekwencji wyszukiwania



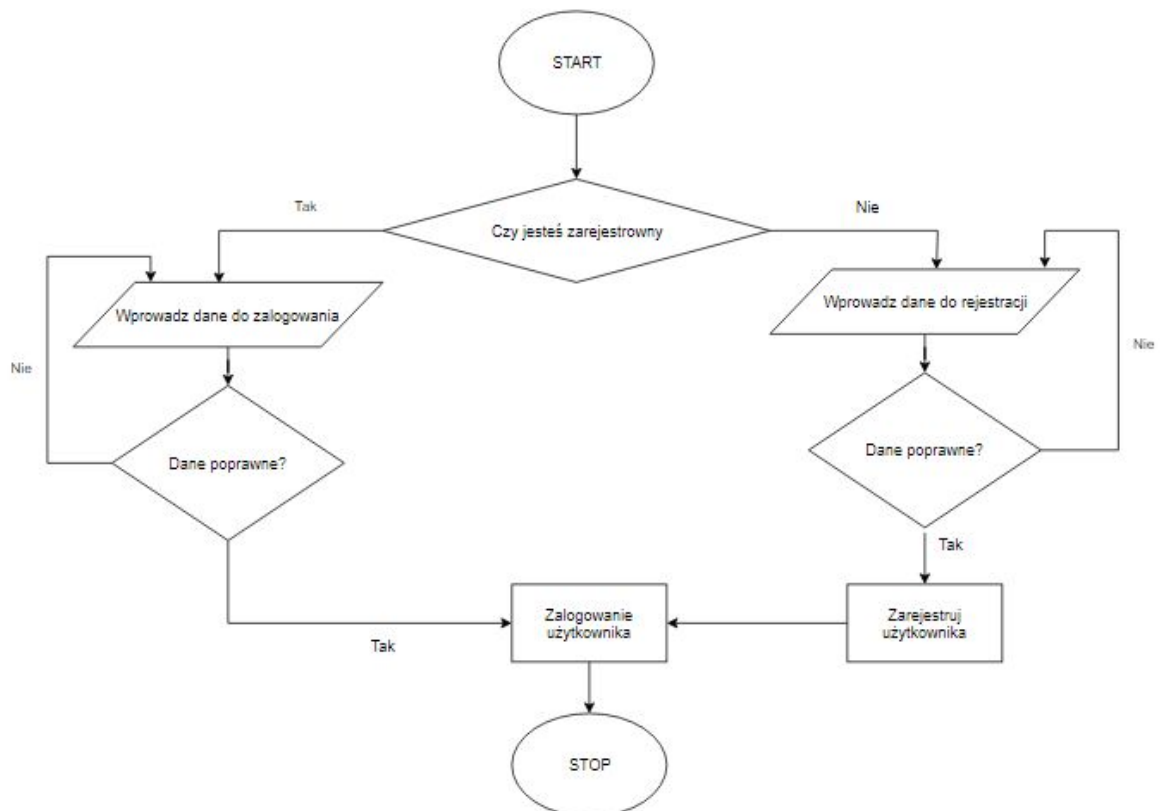
## Diagram dodania Wydarzenia



rys. 12 Diagram sekwencji dodawania wydarzenia

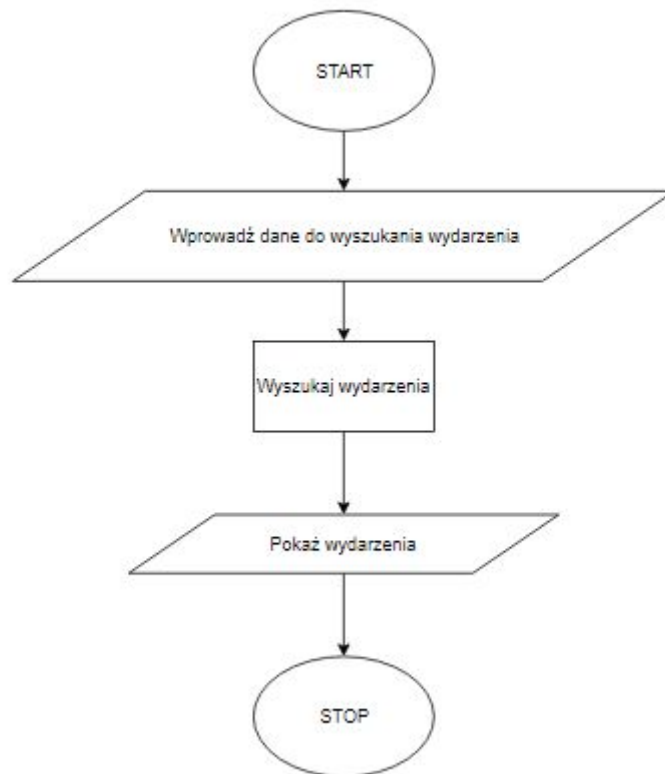
## Diagramy blokowych funkcji

## Rejestracja i logowanie



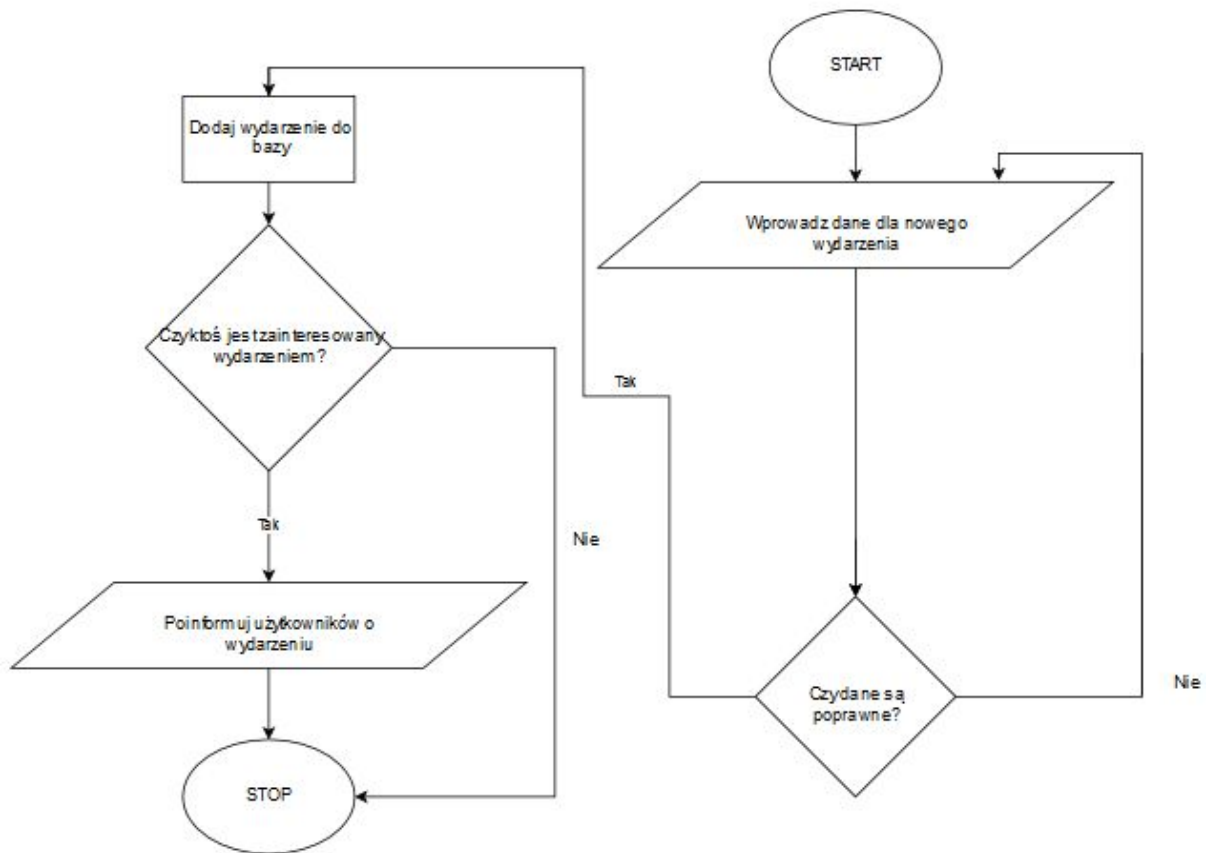
rys.13 Diagram rejestracji i logowania

## Wyszukiwanie wydarzenia



rys.14 Diagram wyszukiwania wydarzenia

## Schemat dodawania wydarzenia



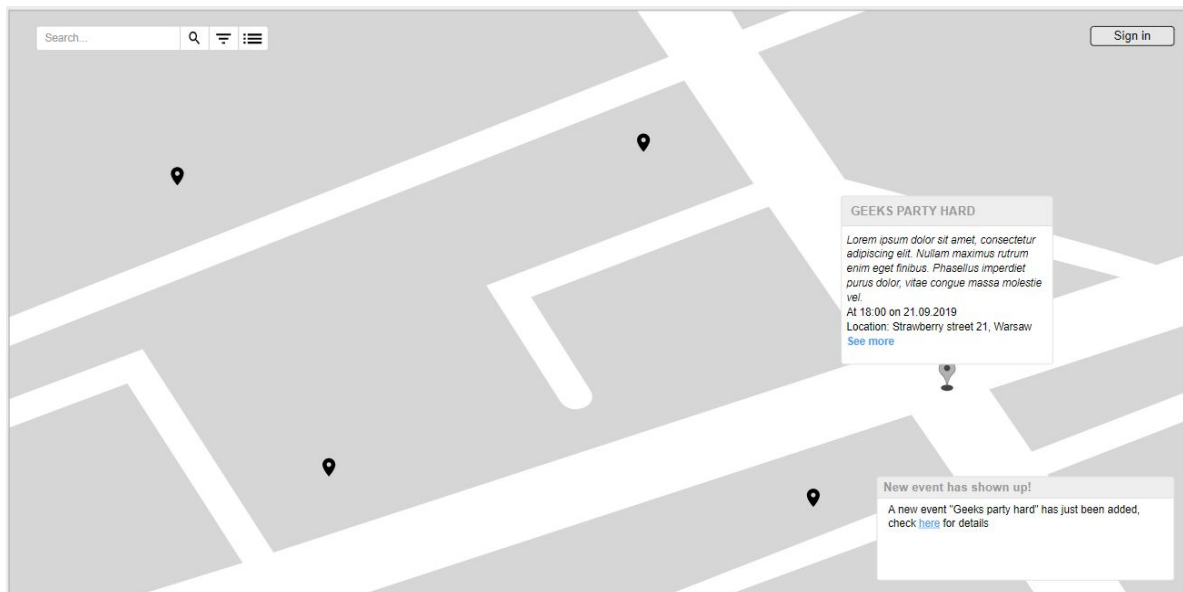
rys.15 Schemat dodawania wydarzenia

## Projekt interfejsu użytkownika

Poniższy podrozdział, w sposób obrazkowy, wraz z krótkim opisem obrazków prezentuje przykładową implementację graficzną interfejsu użytkownika aplikacji. Należy podkreślić, że wszystkie obrazki zawarte w tym podrozdziale są przykładowe i o ile będą one stanowić pewien wzór rozłożenia, może nawet wyglądu elementów na stronie, to wygląd stron może być inny po fazie implementacji.

W pierwszej kolejności omówimy stronę główną.

## Strona główna



rys.16 Strona główna projektu

Zdecydowaną większością na tym widoku będzie mapa, na której będą znaczniki. Znaczniki będą oznaczać miejsca, gdzie odbędą się jakieś wydarzenia. Po kliknięciu w znacznik, powinny pojawić się szczegółowe informacje na temat klikniętego wydarzenia, takie jak:

1. Nazwa wydarzenia
2. Opis wydarzenia
3. Data, czas oraz dokładna lokalizacja wydarzenia
4. Informacja o kosztach wejścia
5. Możliwość przejścia do widoku szczegółowego wydarzenia.

Od dolnej prawej strony ekranu w górę, będą pojawiać się powiadomienia na żywo na temat wydarzeń, które zostały dodane od momentu, kiedy użytkownik ostatnio otworzył stronę. Tego typu powiadomienie powinno zawierać podobny zakres informacji, co w przypadku kliknięcia na znacznik wydarzenia na mapie.

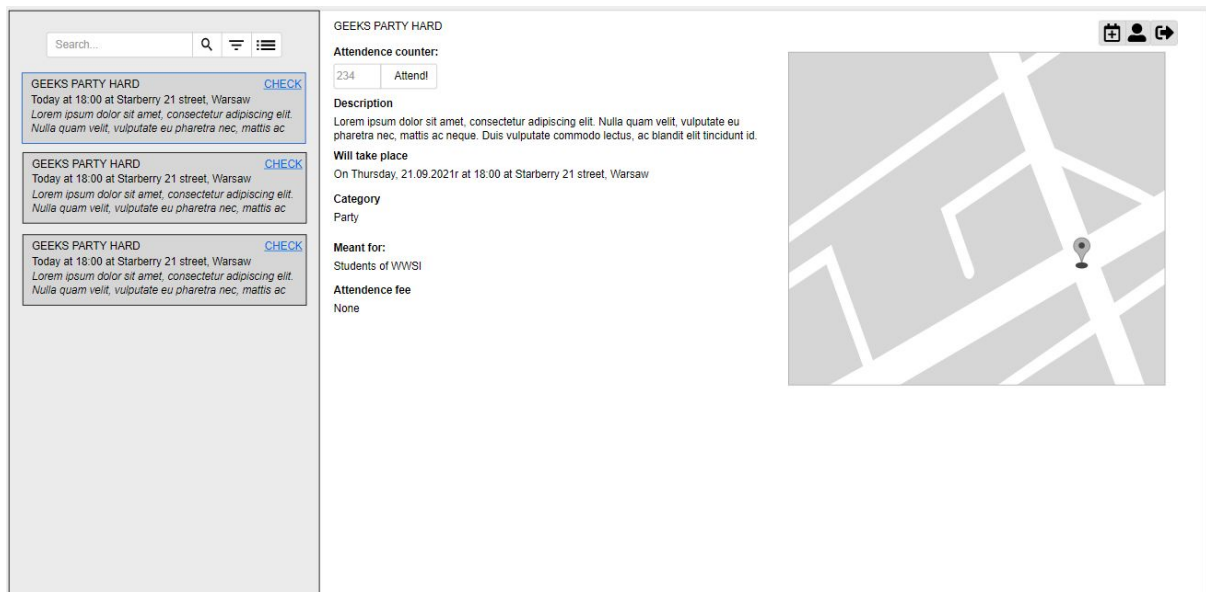
W górnej lewej części strony będzie znajdować się menu nawigacyjne, pozwalające na:

- Wyszukiwanie wydarzenia po nazwie
- Rozwinięcie panelu z dodatkowymi filtrami wydarzeń
- Możliwość przejścia do widoku wydarzeń w postaci listy.

W prawej górnej części strony będzie znajdować się przycisk umożliwiający zalogowanie się. Po kliknięciu w przycisk będzie pojawiać się okno modalne, pozwalające na kontynuowanie procesu logowania.

## Widok listy

Następnie przejdźmy do widoku listy:



rys.17 Widok wydarzeń w postaci listy

W widoku wydarzeń w postaci listy, po lewej stronie znajdować będzie się (kolejno od góry):

- To samo menu pozwalające na wyszukiwanie wydarzeń, co w widoku mapy.
- Lista wydarzeń w lokalizacji.

Po prawej stronie strony natomiast będą znajdować się wszystkie dostępne informacje na temat wybranego z listy wydarzenia.

Te informacje to:

- Nazwa wydarzenia
- Ilość użytkowników wyrażających chęć udziału w wydarzeniu
- Opis wydarzenia
- Data, czas i lokalizacja wydarzenia
- Kategoria wydarzenia
- Kwota pobierana na wejściu wydarzenia

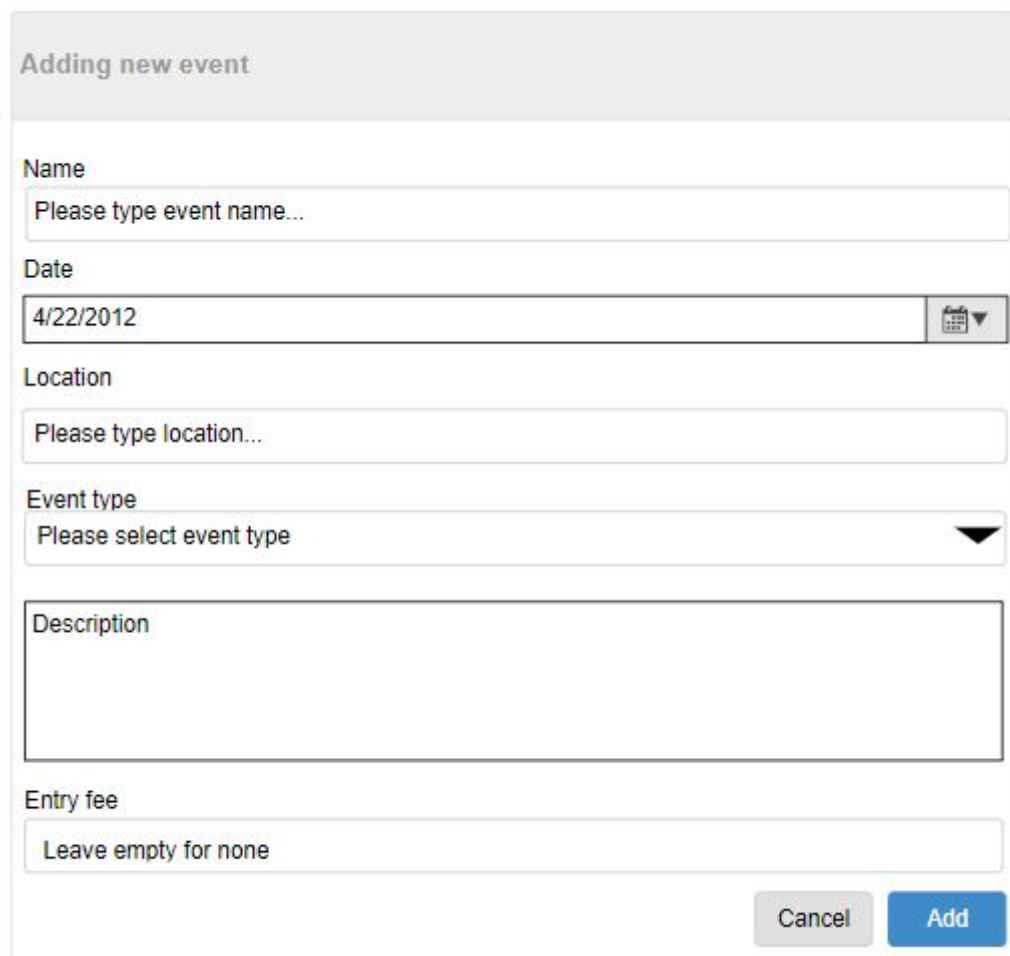
Dodatkowo po prawej stronie od informacji, znajduje się informacyjnie mapa, ze znacznikiem, gdzie wydarzenie się odbędzie.

Powyższy obrazek pokazuje również pasek opcji, wyświetlający się po zalogowaniu użytkownika. Te opcje to:

- Dodawanie wydarzeń, czyli przycisk, po kliknięciu w który, wyskoczy okno modalne, w którym można uzupełnić informacje na temat wydarzenia, a następnie zostanie ono dodane.
- Przycisk - odnośnik do panelu administratora, jeśli oczywiście użytkownik jest administratorem
- Profil użytkownika
- Wylogowanie się

## Dodawanie wydarzenia

Na poniższym obrazku zaprezentowane okno modalne dodawania wydarzenia.



The image shows a modal window titled "Adding new event". It contains several input fields: a text field for "Name" with placeholder "Please type event name...", a date field for "Date" showing "4/22/2012" with a calendar icon, a text field for "Location" with placeholder "Please type location...", a dropdown menu for "Event type" with placeholder "Please select event type", a large text area for "Description", and a text field for "Entry fee" with placeholder "Leave empty for none". At the bottom right, there are two buttons: "Cancel" and "Add".

rys.18 Dodawanie nowego wydarzenia

Okno modalne dodawanie wydarzenia powinno pozwalać na uzupełnienie następujących informacji:

- Nazwa wydarzenia
- Opis wydarzenia
- Data, czas i lokalizacja wydarzenia
- Kategoria wydarzenia
- Grupa docelowa uczestników wydarzenia
- Kwota pobierana na wejściu na wydarzenie

Na dole okno powinny znajdować się dwa przyciski pozwalające na anulowanie dodawania wydarzenia oraz dodanie wydarzenia.

## Okno logowania

Na poniższym obrazku znajduje się przykładowe okno logowania.

The image shows a login form with a light gray header containing the word "Login". Below the header, there are two text input fields. The first is labeled "Email" and contains the placeholder text "Please provide your email". The second is labeled "Password" and contains the placeholder text "Please type your password". Below these fields is a blue hyperlink that reads "Doesn't have an account? Create one here!". At the bottom right of the form are two buttons: a gray "Cancel" button and a blue "Login" button.

rys.19 Logowanie

Okno logowania powinno pozwalać na wpisanie emailu użytkownika oraz hasła do konta. Na dole okna modalnego powinny znaleźć się przyciski do anulowania operacji oraz zalogowania się. Pod polami tekstowymi powinien znajdować się też odnośnik do rejestracji użytkownika.

## Okno rejestracji

Poniższy obrazek prezentuje przykładowe okno rejestracji:



The image shows a registration form titled "Registration". It contains four text input fields with placeholder text: "Please provide your email", "Please type your password", "Please retype your password", and "Please provide your name". At the bottom right, there are two buttons: a grey "Cancel" button and a blue "Register" button.

Registration

Email

Please provide your email

Password

Please type your password

Confirm password

Please retype your password

Name

Please provide your name

Cancel Register

rys.20 Rejestracja

W oknie rejestracji powinny znajdować się pole tekstowe na następujące informacje:

- Email użytkownika
- Hasło, wraz z dodatkowym pole "powtórz hasło", sprawdzające, czy hasła są takie same
- Imię i/lub nazwisko użytkownika

Na dole okna rejestracji powinny znajdować się przyciski do anulowania i kontynuowania operacji.

## Profil użytkownika

Na poniższym obrazku przedstawiony został przykładowy profil użytkownika:

rys.21 Profil użytkownika

Profil użytkownika powinien pozwolić użytkownikowi na:

- Przeglądanie oraz modyfikację danych o własnym profilu
- Przeglądanie wydarzeń, do których się zapisał
- Przeglądanie własnych, stworzonych wydarzeń
- Usunięcie konta

## Panel Admina

Na poniższym obrazku znajduje się przykładowy panel administratora.

rys.22 Panel administratora

W panelu administratora będą znajdować się odnośniki do zarządzania poszczególnymi obiektami, w tym wypadku są to: wydarzenia oraz użytkownicy.

Po przejściu do odpowiedniej kategorii obiektów, zostanie wyświetlona tabela, która pozwoli na edycję poszczególnego rekordu lub jego usunięcie.

# Projekt testów kontrolnych

## Scenariusze testowe:

### Scenariusz nr 1. Rejestracja nowego użytkownika

#### Kroki:

Wejść na stronę [www.eventnotifier.dev](http://www.eventnotifier.dev)

Nacisnąć przycisk Sign In

Następnie kliknąć w link pod tytułem “Doesn’t have an account? Create one here!”

Wprowadzić prawidłowe email, password, name, np., „[test@example.com](mailto:test@example.com), pass, Test”,

Kliknąć Register.

#### Oczekiwany wynik:

1. Okno z informacją o poprawnym zarejestrowaniu nowego użytkownika
2. Przekierowanie na stronę główną po 5 sekundach od wyświetlenie komunikatu

### Scenariusz nr 2. Dodawanie nowego wydarzenia (Eventu)

#### Kroki:

Wejść na stronę [www.eventnotifier.dev](http://www.eventnotifier.dev)

Nacisnąć przycisk Sign In

Zalogować się na konto testowego użytkownika (login — test, hasło — test),

Wejść ponownie na stronę główną i kliknąć przycisk dodania nowego wydarzenia

Wprowadzić prawidłowe dane do: Name, Date, Location, Event type, Description, Entry fee

Kliknąć Add.

#### Oczekiwany wynik:

1. Modal z formularzem dodawania nowego wydarzenia zamyka się i wyświetla się lista, w której jest obecny nowy event
2. Na nowe wydarzenie można kliknąć
3. W otwartej karcie można zobaczyć informacje dotyczące nowego wydarzenia

### Scenariusz nr 3. Zapisanie się na wydarzenie

#### Kroki:

Wejść na stronę [www.eventnotifier.dev](http://www.eventnotifier.dev)

Nacisnąć przycisk wyboru listy, znajdujący się w lewym-górnym rogu

Nacisnąć na pierwszy kafelek z widocznej listy wydarzeń

Nacisnąć przycisk “Attend”

**Oczekiwany wynik:**

1. Komunikat informujący o przebiegu operacji (Ścieżka pozytywna, bądź negatywna)

**Scenariusz nr 4. Wypisanie się z wydarzenia****Kroki:**

Wejść na stronę [www.eventnotifier.dev](http://www.eventnotifier.dev)

Nacisnąć przycisk wyboru listy, znajdujący się w lewym-górnym rogu

Nacisnąć na pierwszy kafelek z widocznej listy wydarzeń

Nacisnąć przycisk "Resigns"

**Oczekiwany wynik:**

1. Komunikat informujący o przebiegu operacji (Ścieżka pozytywna, bądź negatywna)

## Testy obciążeniowe

Testy obciążeniowe aplikacji zostaną wykonane przy pomocy darmowej aplikacji JMeter. Jest to narzędzie stworzone w języku Java. JMeter komunikuje się z testowanym serwisem, tak jakby wykonywał to prawdziwy użytkownik. Działa na zasadzie wysyłania Request'ów i sprawdzania Respons'ów zwracanych przez serwer aplikacji.

**Założenia:**

- Pula użytkowników: 2000
- Site [www.eventnotifier.dev](http://www.eventnotifier.dev)
- Port: 80

Po podstawowej konfiguracji programu JMeter, należy przygotować zapytania wg scenariuszy. Strony, które zostaną objęte testami obciążeniowymi:

- Strona Logowania
- Strona Rejestracji
- Strona Główna
- Proces zapisywania się na wydarzenie

## Testy jednostkowe

Każdy test będzie testować niewielki fragment systemu, aby w przypadku jego negatywnego wyniku można było łatwo odnaleźć miejsce błędu. Kolejnym ważnym elementem testów, będzie ich szybkość wykonywania, im szybciej tym lepiej. Dzięki zastosowaniu w.w reguł możemy uzyskać najlepsze rezultaty z stworzonych testów. Testy będą też tworzone dla warunków brzegowych, jest to jeden z najwartościowszych aspektów testowania, ponieważ pozwala wykryć i przetestować obszary, w których prawdopodobieństwo niepoprawnego działania kodu jest największe.

Narzędziem, które posłuży do testowanie kodu będzie PHPUNIT. Narzędzie to oferuje nam kilka przydatnych funkcji takich jak:

- Interfejs linii komand do wykonywania testów
- Data Providers - generator zestawów danych
- Testowanie kodu z użyciem bazy danych
- Testowanie wyjątków
- Wsparcie dla mock objects
- Generator raportów pokrycia kodu

Przykładowe klasy, które zostaną poddane testom:

Nazwa klasy	Założenia
UserRepository	Test ma za zadanie sprawdzić możliwość rejestracji użytkowników, oraz sprawdzenia mechanizmu logowania w brzegowych warunkach. Test będzie zakładał ścieżki pozytywne, oraz negatywne.
EventRepository	Test ma za zadanie przetestować możliwość tworzenia nowych wydarzeń, edycji bieżących wydarzeń oraz usuwania już istniejących. Dodatkowy testy będą sprawdzały uprawnienia użytkowników
EventMemberRepository	Test będzie zawierał ścieżki do zapisania się na wydarzenia, oraz odwołanie swojego udziału.

## ETAP III

Implementacja bazy danych

PRZEDSTAWIENIE SPOSOBU IMPLEMENTACJI WYBRANYCH ELEMENTÓW BAZY DANYCH - STRUKTURY ORAZ LOGIKI

Implementacja warstwy logicznej

PRZEDSTAWIENIE ROZWIĄZAŃ PROGRAMISTYCZNYCH DLA WYBRANYCH PROBLEMÓW(STRUKTUR, ALGORYTMÓW) OPISANIE SPOSOBU REALIZACJI WYBRANEGO ALGORYTMU

Implementacja GUI

PRZEDSTAWIENIE WIDOKU UTWORZONYCH ELEMENTÓW GUI (strony, podstrony, itp.)

Gotowy system informatyczny

PRZEKAZANIE KODÓW ŹRÓDŁOWYCH WSZYSTKICH WARSTW SYSTEMU (Baza danych, logika interfejs)

# ETAP IV

## Wyniki przeprowadzonych testów

- WYKONANIE TESTÓW ILOŚCIOWYCH/OBCIĄŻENIOWYCH Z WYKORZYSTANIEM DOSTĘPNYCH NARZĘDZI
- WYKONANIE TESTÓW SPRAWDZAJĄCYCH DLA POSZCZEGÓLNYCH MODUŁÓW SYSTEMU