**Uniwersytet Technologiczno –Przyrodniczy w Bydgoszczy**

Wydział Telekomunikacji, Informatyki i Elektrotechniki



Projekt

Programowanie aplikacji internetowych

**Dokumentacja projektu**

**„Serwis umożliwiający organizację imprez”**

***COVID AFTER-PARTY***

*Michael Szychulski,*

*Nr indeksu: 113115*

*Dominik Wiśniewski,*

*Nr indeksu: 113131*

*Michał Sulecki,*

*Nr indeksu: 113112*

*Informatyka stosowana*

*Semestr VI*

*grupa 2   
Link do repozytorium projektu:* <https://github.com/MCtyskie/pai_project>

Spis treści

[1. Specyfikacja projektu 2](#_Toc75205776)

[a. Ogólny opis 2](#_Toc75205777)

[b. Cel i przeznaczenie 3](#_Toc75205778)

[c. Specyfikacja wymagań na oprogramowanie 3](#_Toc75205779)

[a) Wymagania funkcjonalne: 3](#_Toc75205780)

[b) Wymagania bazy danych: 3](#_Toc75205781)

[c) Wymagania bezpieczeństwa: 3](#_Toc75205782)

[d) Wymagania sprzętowo-wydajnościowe: 4](#_Toc75205783)

[e) Wymagania na testowanie: 4](#_Toc75205784)

[d. Architektura oprogramowania 4](#_Toc75205785)

[a) Diagram architektury 5](#_Toc75205786)

[b) Opis elementów składowych oprogramowania 5](#_Toc75205787)

[c) Alokacja wymagań w architekturze 6](#_Toc75205788)

[e. Implementacja architektury 6](#_Toc75205789)

[a) Admin Service 7](#_Toc75205790)

[b) Login Service 7](#_Toc75205791)

[c) Event Service 8](#_Toc75205792)

[d) User Service 12](#_Toc75205793)

[e) Review Service 13](#_Toc75205794)

[f) Invitation Service 13](#_Toc75205795)

# Specyfikacja projektu

## Ogólny opis

System z założenia ma umożliwić interakcje pomiędzy organizatorem wydarzenia, a uczestnikami. Organizator tworzy nowe wydarzenie, określa liczbę gości, miejsce, termin i pozostałe szczegóły wydarzenia. Po ustaleniu szczegółów organizator może opublikować wydarzenie, aby było dostępne dla uczestników. Założyciel może dodatkowo wysyłać zaproszenia do grupy jego np. wcześniejszych uczestników, za pośrednictwem e-maili. Uczestnicy imprez po wejściu na aplikacje maja dostępną wyszukiwarkę okolicznych wydarzeń. Dostępne jest wyszukiwanie interesujących wydarzeń poprzez system tagów, które określają tematykę. Następnie po wybraniu wydarzenia można przejrzeć szczegóły, które zostały ustalone przez organizatora i zarezerwować swoje miejsce na wydarzeniu, lub określić swoje uczestnictwo.

Poprzez listę dostępnych wydarzeń i możliwość rezerwacji miejsca, założyciel wydarzenia od razu wie, ile gości może się spodziewać i jakie środki należy przygotować. Użytkownicy od razu mają informację, ile miejsc jest jeszcze dostępnych i mogą łatwiej zapisać się na dane wydarzenie. Dodatkowo aplikacja przewiduje wyszukiwanie wydarzeń wedle tematyki co znacząco ułatwi wyszukiwanie interesujących wydarzeń.

## Cel i przeznaczenie

Projekt ma umożliwić organizację wydarzeń zbiorowych poprzez poprawę widoczności, przystępne i szybkie zakładania, publikację i zarządzanie wydarzeniami i listą uczestników .

Przeznaczeniem projektu jest implementacja aplikacji na chmurze obliczeniowej, do której można będzie uzyskać dostęp przez przeglądarkę. Aplikacja ma za zadanie umożliwić przejrzysty, szybki i przystępny sposób na zakładanie i zarządzanie wydarzeniami zbiorowymi i listą uczestników, oraz na przeglądanie i wyrażanie uczestnictwa w opublikowanych wydarzeniach. Aplikacja jest skierowana do każdej grupy wiekowej

## Specyfikacja wymagań na oprogramowanie

### Wymagania funkcjonalne:

#### Rejestracja użytkownika

* + - 1. Imię – pole tekstowe \*
      2. Nazwisko – pole tekstowe \*
      3. Adres Email – pole tekstowe
      4. Hasło – pole tekstowe \*
      5. Data urodzenia – data
      6. Miasto zamieszkania – pole tekstowe \*
      7. Numer telefonu – pole tekstowe \*

#### Logowanie użytkownika za pomocą email i hasła

#### Tworzenie i zarządzanie wydarzeniami

#### Dołączanie do wydarzeń przez użytkownika

#### Zarządzanie zaproszeniami jako organizator imprezy

#### Wyszukiwanie wydarzeń za pomocą:

#### Zarządzanie komentarzami dla zakończonych wydarzeń, w których użytkownik wziął udział

### Wymagania bazy danych:

Relacyjna baza danych oparta na mySQL. Baza będzie posiadała jednego użytkownika z pełnymi uprawnieniami. Dostęp do niego będzie można uzyskać za pomocą panelu administratora aplikacji.

### Wymagania bezpieczeństwa:

Bezpieczeństwo jest realizowane za pomocą integracji Spring Security z systemem tokenów realizowanym za pomocą JWT. Użytkownik musi zarejestrować swoje konto, następnie się zalogować z poprawnymi danymi i wtedy otrzymuje token autoryzujący, który umożliwia poruszanie się i korzystanie po serwisie. Token zostaje otrzymany na określony czas. Akcje użytkownika odświeżają czas wygaśnięcia. Po minięciu określonego czasu bez żadnej akcji użytkownika, zostaje on wylogowany.

Role użytkownika dzielą się na administratorów i zwykłych użytkowników. Administratorzy mają pełen dostęp do każdych funkcjonalności, a użytkownicy posiadają dostęp do wszystkich innych, poza panelem administratorskim. Mogą oni edytować tylko swoje wydarzenia, zaproszenia czy też komentarze, gdzie admin ma dostęp do edycji jakichkolwiek wydarzeń, zaproszeń etc.

### Wymagania sprzętowo-wydajnościowe:

Jednostka obliczeniowa z dostępem do Internetu na której uruchomiony będzie aplikacja obsługująca klientów i bazę danych.

Minimalne wymagania serwera:

1 TB pojemności dyskowej

16 GB RAM

Procesor 6-cio rdzeniowy, 12-to wątkowy

Przepustowość sieciowa powyżej 10Mbps pobierania i wysyłania

### Wymagania na testowanie:

Przeprowadzone zostaną testy jednostkowe sprawdzającą logikę wykonywaną na danym endpoincie. Pokrycie kodu zakładane jest na powyżej 50%. Planowane jest kilka większych testów funkcjonalnych i integracyjnych, które sprawdzą współpracę poszczególnych części systemu. Będzie to sprawdzenie kluczowych i bazowych funkcjonalności, gdyż większa ilość takich testów może przełożyć się negatywnie na wydajność możliwego użycia pipeline’a (rozszerzenie funkcjonalności wchodzi na główną gałąź projektu po zapewnieniu, że wszystkie poprzednie testy działają). Kluczowe funkcjonalności:

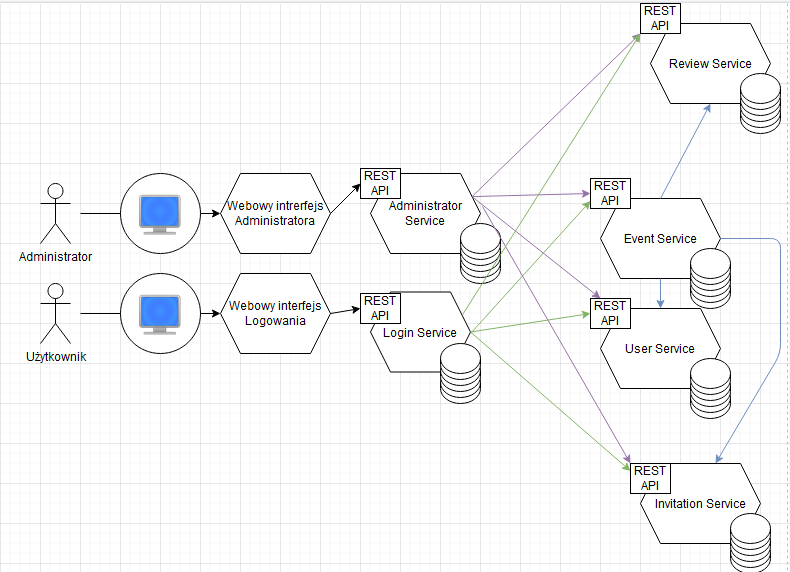
* Tworzenie i zarządzanie wydarzeniami
* Wysyłanie, odbieranie, akceptowanie i odrzucanie zaproszeń
* Rejestracja użytkownika
* Logowanie użytkownika

## Architektura oprogramowania

Projekt jest wykonany jako aplikacja sieciowa, gdzie:

* Backend – część serwerowa – oparta na języku Java z wykorzystaniem frameworka Spring Boot
* Frontend – część kliencka – oparta na Javascript z wykorzystaniem frameworka React.js i pomniejszych pakietów stylizacji np. Material-UI, Bootstrap 4.

### Diagram architektury



Aby dobrze zrozumieć powyższą architekturę należy mieć na uwadze, że cała aplikacja wyszczególnia jeden typ użytkownika, który może być zarówno organizatorem wydarzenia, jak i uczestnikiem innych wydarzeń, z tego wynika, że może on osobno dostać się na menu zarządzania zaproszeniami od strony bycia uczestnikiem, jak i organizatorem i ta sama formuła aplikowana jest we wszystkich serwisach widocznych powyżej. Szczegóły tego rozdziału zostaną opisane w implementacji oprogramowania.

Głównym serwisem jest oczywiście serwis eventów, który łączy ze sobą pewne zależności zaproszeń i ocen dla danych wydarzeń i korzysta z serwisu user-ów, aby rozróżnić, czy ma do czynienia z organizatorem, bądź też uczestnikiem.

### Opis elementów składowych oprogramowania

* 1. Administrator Service określa serwis udostępniany administratorom, który służy za bramę do wykonywania wszystkich możliwych czynności na innych serwisach
  2. Login Service jest to element służący do autoryzowania użytkowników wewnątrz aplikacji
  3. Event Service zawiera główną funkcjonalność zarządzania wydarzeniami //do dopisania
  4. User Service wykorzystywany jest w celu zarządzania wszystkimi możliwymi funkcjonalnościami użytkownika
  5. Review Service dostępny jest z poziomu Eventu, gdzie oferuje funkcjonalności związanie z wyświetlaniem ocen i komentarzy, a także z poziomu User Service, za którego pomocą mogą być wystawiane takowe komentarze
  6. Invitation Service określa z poziomu Eventów możliwość wysyłania zaproszeń dla użytkowników, którzy z poziomu swojego serwisu mogą je zaakceptować lub odrzucić.

### Alokacja wymagań w architekturze

Odniesienie się do punktów z [Opis elementów składowych oprogramowania](#_Opis_elementów_składowych) do funkcjonalności opisanych w [Wymagania funkcjonalne:](#_Wymagania_funkcjonalne:)

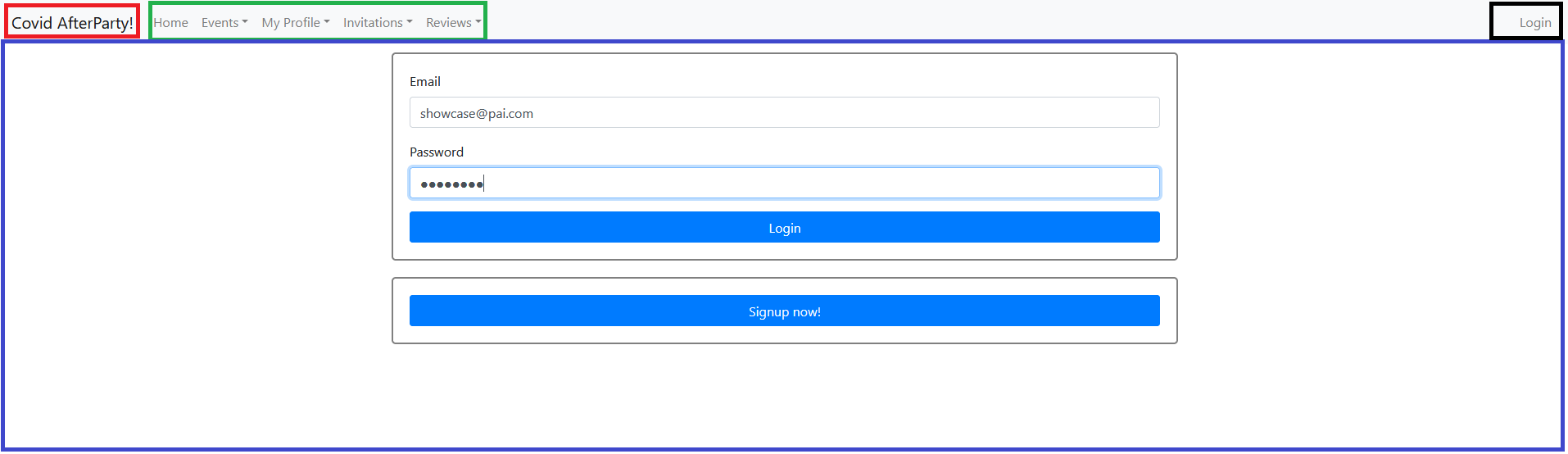
1. Administrator Service

Realizuje wszystkie możliwe funkcjonalności systemu poza rejestracją – musi być ustanowiony bezpośrednio w bazie danych/zarejestrowany specjalną metodą na serwerze.

1. Login Service
   1. Rejestracja użytkownika z danymi polami
   2. Logowanie za pomocą email i hasła
2. Event Service
   1. Organizator wydarzeń:
      1. Tworzenie i zarządzanie wydarzeniami
      2. Pobranie listy organizowanych wydarzeń
      3. Edycja organizowanych wydarzeń
   2. Uczestnik wydarzeń:
      1. Wyszukiwanie wydarzeń i pobranie dostępnych wydarzeń
      2. Przeglądanie szczegółów danego wydarzenia – listy zaproszonych, ocen itp.
3. User Service
   1. Pobranie listy użytkowników – admin panel
   2. Pobranie informacji danego użytkownika
   3. Edycja danych konta użytkownika
4. Review Service
   1. Zarządzanie komentarzami dla zakończonych wydarzeń, w których użytkownik wziął udział
   2. Pobranie listy wystawionych komentarzy
5. Invitation Service
   1. Organizator wydarzeń:
      1. Akceptacja/Odrzucenie prośby o dołączenie do imprezy przez uczestników
   2. Uczestnik wydarzeń
      1. Dołączanie do wydarzeń poprzez wysłanie zaproszenia
      2. Wyświetlanie aktualnych zaproszeń i ich statusów

## Implementacja architektury

Opis implementacji będzie głównie bazował na poszczególnych modułach dostępnych poprzez warstwę kliencką aplikacji z odniesieniami do realizacji po stronie serwerowej.

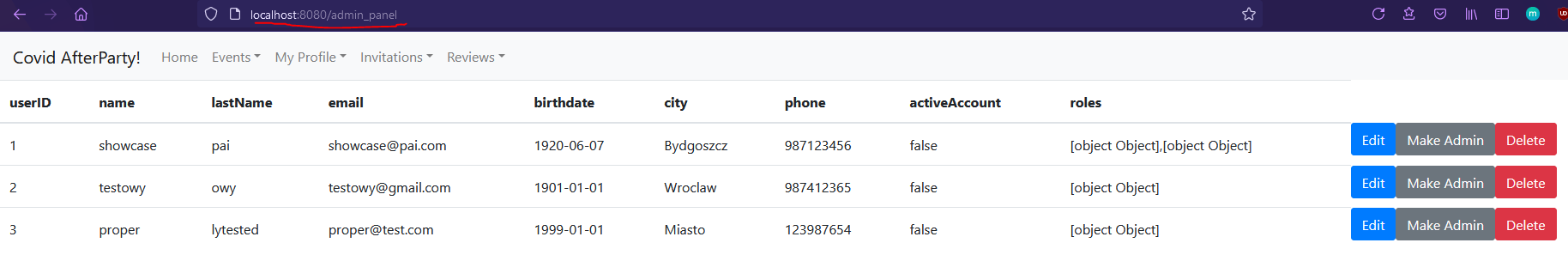


Główny widok aplikacji składa się z 4 sekcji:

* Kolor czerwony – Tytuł aplikacji
* Kolor zielony – Navbar – pasek nawigacji poszczególnych modułów, do których użytkownik ma dostęp po prawidłowym zalogowaniu
* Kolor czarny – Przycisk logowania/wylogowania – zmienia status po zalogowaniu/wylogowaniu
* Kolor niebieski – kontent strony, danego modułu

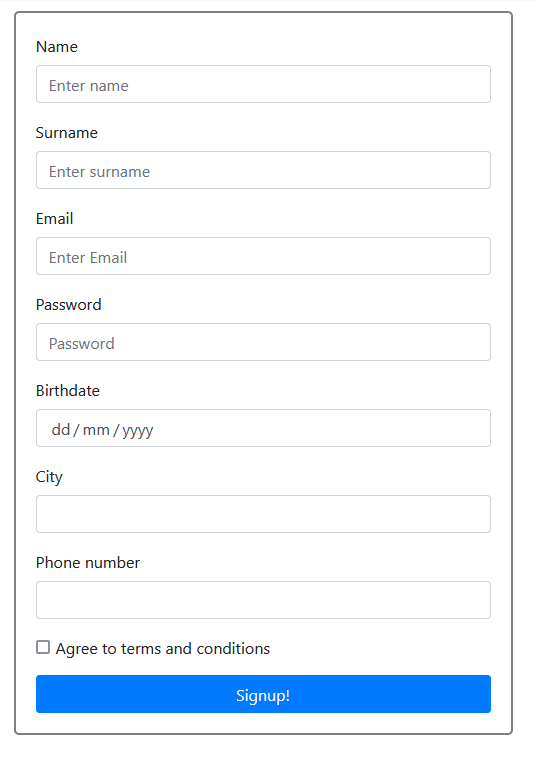
### Admin Service

Wejście na serwis admina odbywa się poprze zalogowania opisanego w punkcie poniżej - [Login Service](#_Login_Service) – i cały dostęp do tego modułu jest sprawdzany po stronie serwerowej poprzez adnotację Pre Authorize, która jest filtrem przepuszczającym użytkownika na podstawie zapisanej roli. We frontendzie jest on ukryty pod [www.adres\_serwera:port/admin\_panel](http://www.adres_serwera:port/admin_panel) i nie ma do niego bezpośredniego przejścia w modułach poza wpisaniem adresu na „sztywno”. Głównym zastosowaniem tego panelu jest pobranie listy użytkowników, możliwość ich edycji i usuwania, lub też nadawania praw administratorskich:

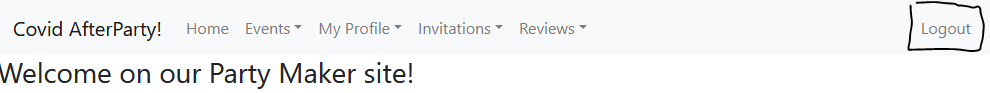


Jeżeli chodzi o dostęp do reszty serwisów tak jak było to opisywane w specyfikacji, administrator może np. usunąć wydarzenie nie będąc jego organizatorem, usunąć dane zaproszenie, zmienić manualnie status wydarzenia.

### Login Service

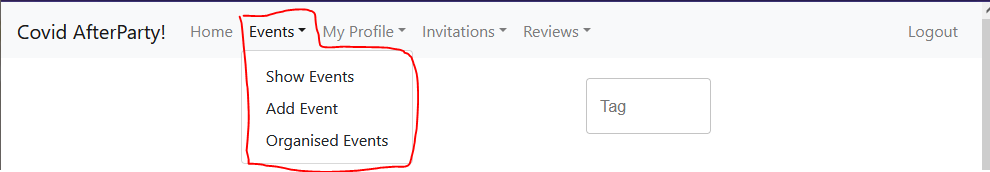
Poprzez wejście na sekcję czarną – moduł logowania – widać formularz logowania i przycisk do rejestracji konta widoczny na zrzucie ekranu powyżej. Fomularz rejestracyjny realizujący tworzenie konta użytkownika w serwisie wygląda następująco:  


Po udanym zalogowaniu użytkownik jest przeniesiony na stronę główną z zmienionym statusem bloku czarnego na logout:



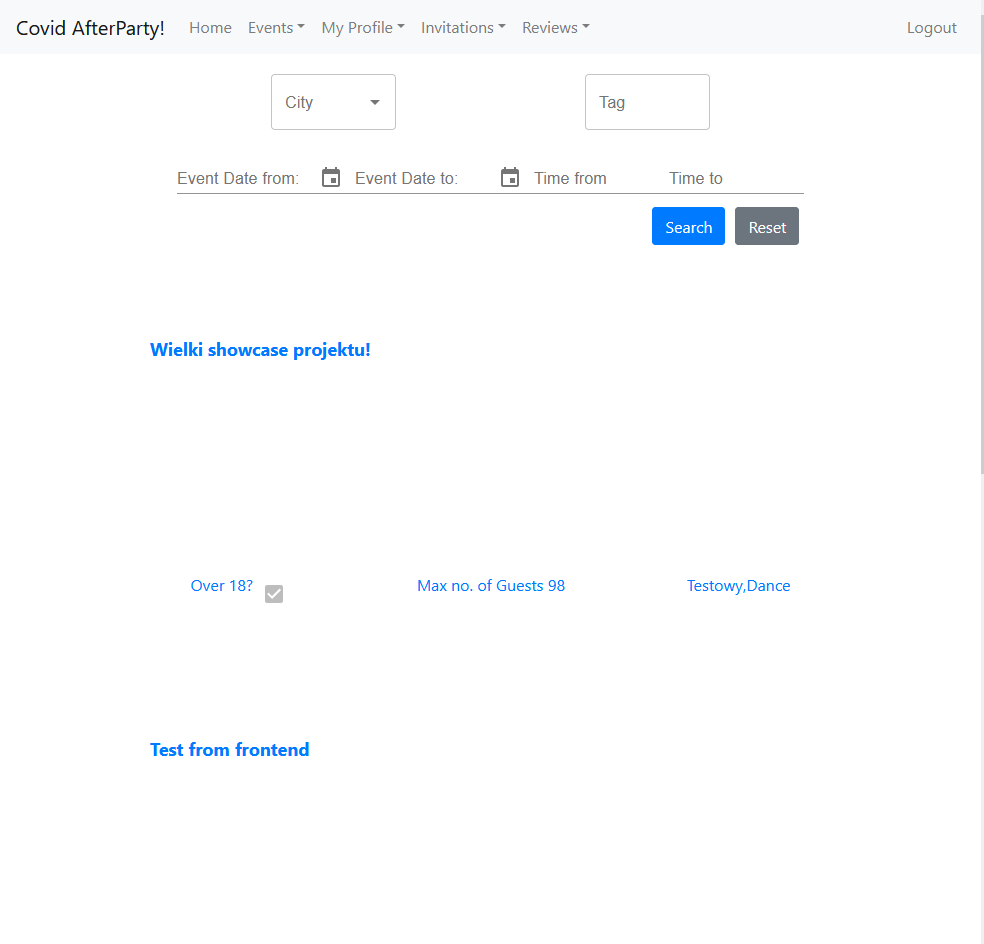
### Event Service

Największy moduł całej aplikacji, kreowany na wzór wyglądu aplikacji mobilnych, aby zwiększyć możliwości przyszłego rozwinięcia systemu o tą platformę. Sugerowany design bazuje na „wall-u” facebooka, gdzie każde wydarzenie jest widoczne i posiada główną cześć ekranu, a lista jest przewijana i „doładowuje” kolejne wydarzenia po przekroczeniu paginacji.

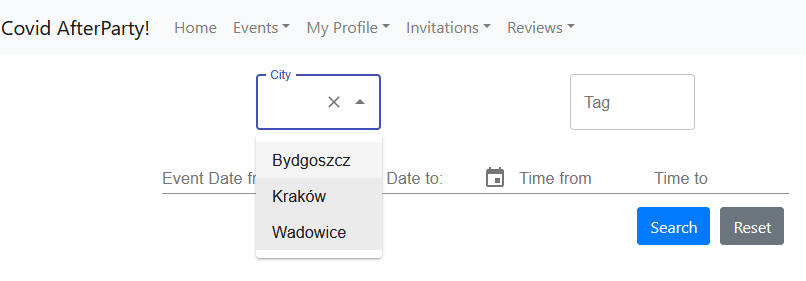


Dostępne menu to:

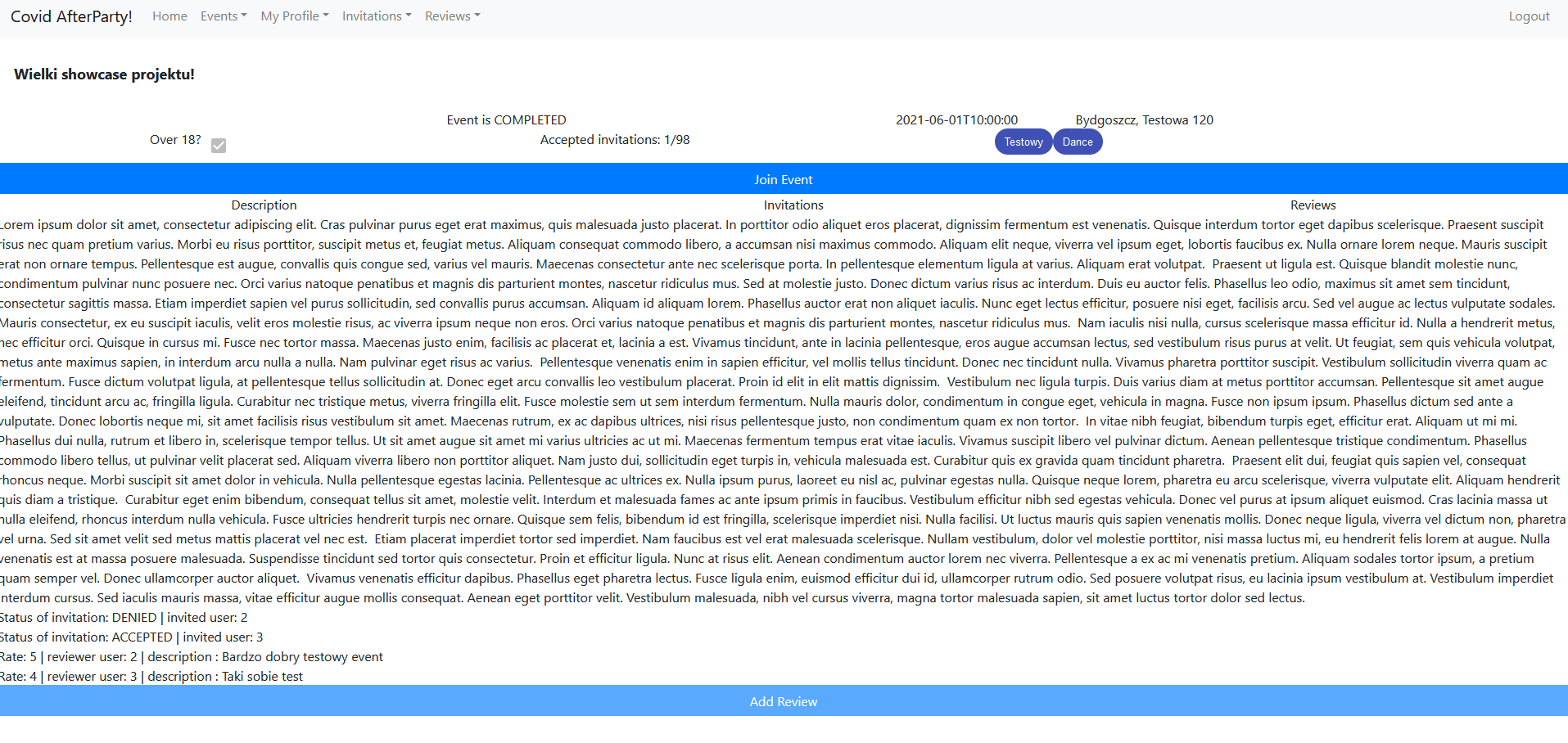
* Show Events – pobranie listy eventów I wyświetlenie ich w sekcji wraz z dostępnymi filtrami



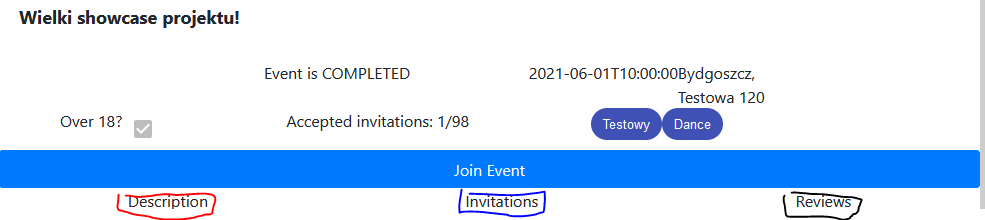
Jak widać na powyższym zrzucie ekranu, w górnej sekcji modułu znajduje się 6 filtrów:

* + City -> Wybór filtrowania po mieście, gdzie lista rozwijana jest zapełniania z serwera po unikalnych nazwach miast, wraz z funkcjonalnością autouzupełniania:  
    
  + Tag -> użytkownik może wpisać jedno dowolne słowo kluczowe, które potem jest parsowane po części serwerowej na podstawie działania LIKE %tag%.
  + Event Date from/to -> określenie ram datowych odbycia się wydarzenia
  + Time from/to -> określenie ram czasowych odbycia się wydarzenia; działa to na zasadzie, gdy wybierzemy filtr dat od 2020-06-20 do 2020-06-22 i filtr czasowy od 19 do 22, to zakres filtracji odbędzie się wtedy w ramach <2020-06-20 19:00; 2020-06-22 22:00>.

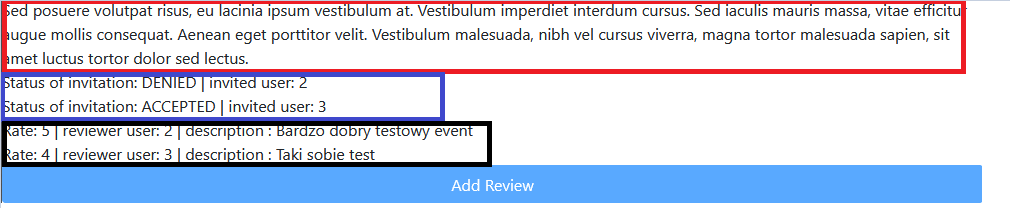
Po kliknięciu w wybrane wydarzenie następuje przejście do widoku szczegółów:



W widoku szczegółów znajduje się pomocniczy pasek nawigacyjny z trzema sekcjami, gdzie zaimplementowano funkcję odnośnika do sekcji treści w widoku; po kliknięciu np. w sekcje Reviews, zastosowane zostanie przejście poprzez scroll do tej sekcji:



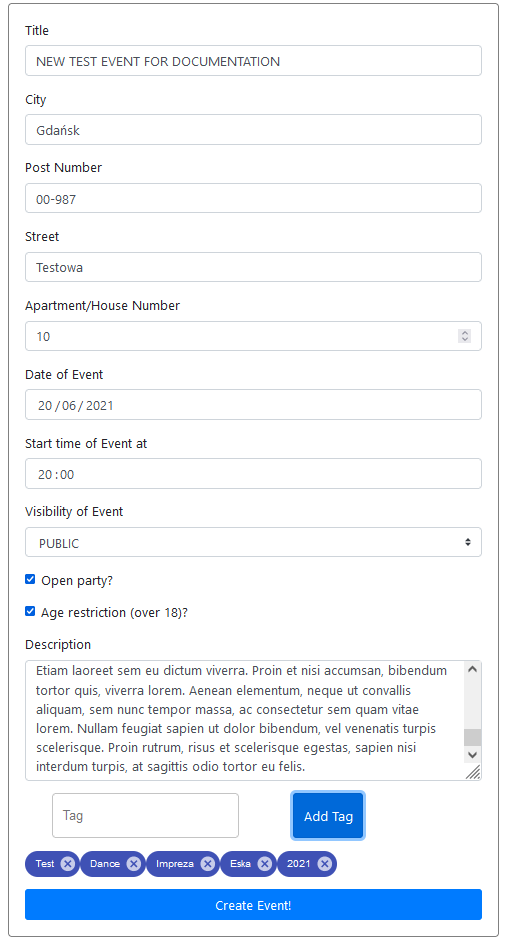
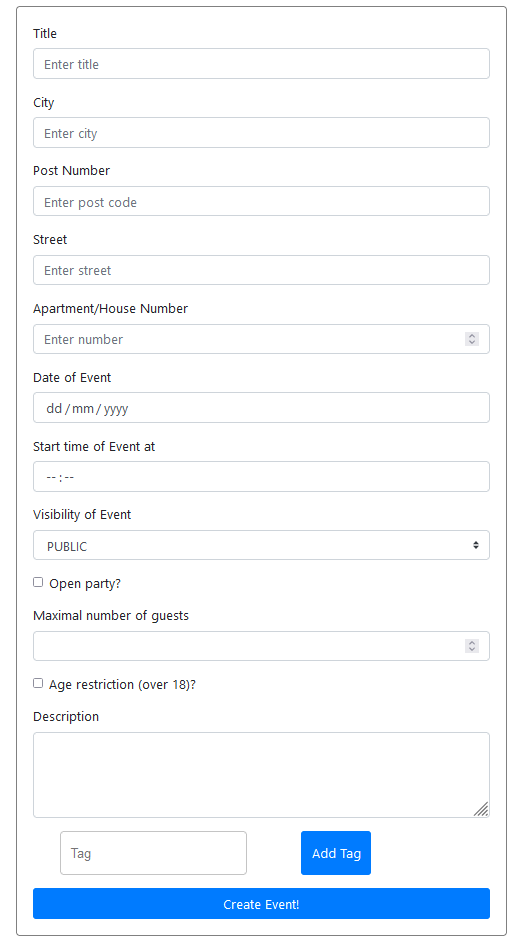
Wygląd poniższych sekcji z oznaczeniem kolorystycznym:



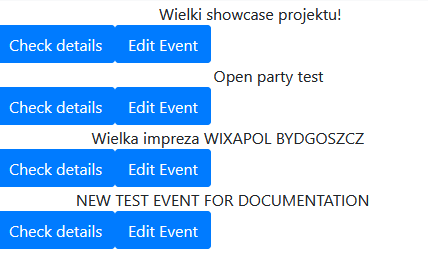
Poprzez stronę serwerową w tym widoku pobierane są listy zaproszeń oraz recenzje dla tego eventu. Wraz z tymi szczegółami, dodatkowo jest sprawdzany parametr, czy ten użytkownik może dodać recenzję, gdzie warunki opisane są w punkcie [Review Service](#_Review_Service).

Po części serwerowej również ustawiony jest scheduler, który cyklicznie co minutę(ze względu na najmniejszą jednostkę czasową rozpoczęcia wydarzenia) sprawdza aktualną datę z datą rozpoczęcia wydarzenia i aktualizuje jego status na COMPLETED.

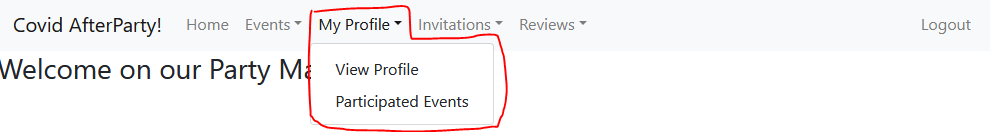
* Add Event – formualrz dodania eventów przez użytkownika, który zostaje organizatorem tego właśnie wydarzenia. Organizator musi wpisać wszystkie pola obowiązkowe. Pola tagów są nieobowiązkowe i dodawane są pojedynczo. Jeżeli zostanie zaznaczone okno z „Open Party?” to usunięte zostanie pole do wpisania maksymalnej liczby gości. Przykładowa realizacja poniżej:



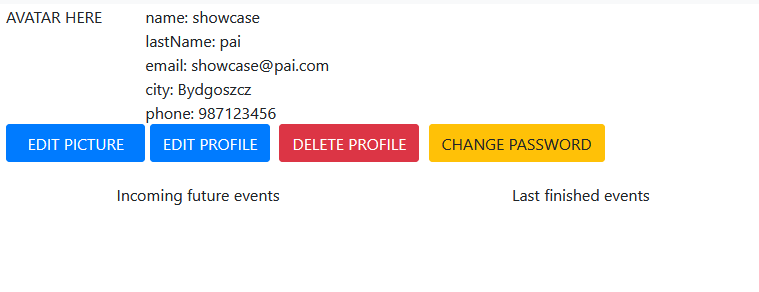
* Organised Events – pobranie listy eventów, w których użytkownik jest organizatorem. Dodatkowo dodano możliwość edycji eventu poprzez „popup” – wyskakujący formularz i przycisk z łączem do szczegółów eventu.

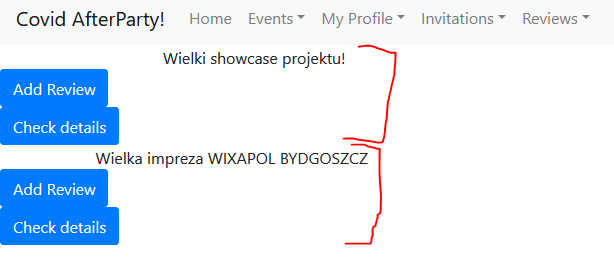


### User Service

Sekcja ta znajduje się w zakładce My Profile, gdzie znajdują się dwie zakładki:  


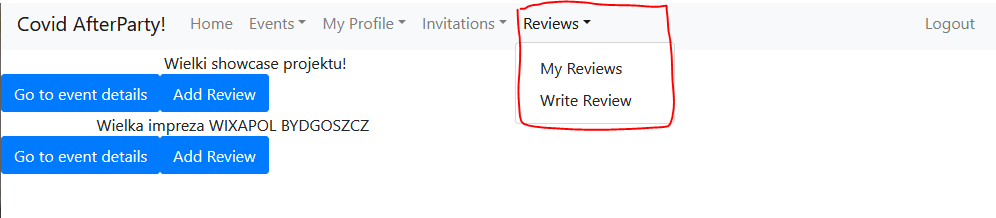
* View profile – realizująca pobranie i możliwość edycji danych własnego profilu, dodatkowo wyświetlane są najbliższe 5 eventów pod względem daty i 5 zakończonych od najnowszych wydarzeń, w których użytkownik wziął udział.



* Participated Events – pobranie listy eventów, w których użytkownik brał udział i możliwość przejścia do szczegółów, lub dodanie recenzji:  
  

### Review Service

Realizacja znajduje siew module Reviews, gdzie dostępne są opcje:



* My Reviews – pobranie listy recenzji wystawionych przez użytkownika
* Write Review – pobranie listy wydarzeń, do których użytkownik może, ale jeszcze nie wystawił żadnego komentarza.

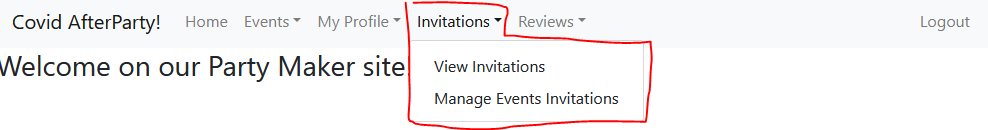
Należy pamiętać, iż jeden użytkownik może napisać tylko jeden komentarz do danego wydarzenia, po czym blokowana jest ta możliwość i sprawdzana specjalnym parametrem po części serwerowej w tym serwisie.

Serwer posiada metodę, która sprawdza dla danego użytkownika czy może on wystawić komentarz/recenzję. Użytkownik może dodać nową recenzję po spełnieniu danych warunków:

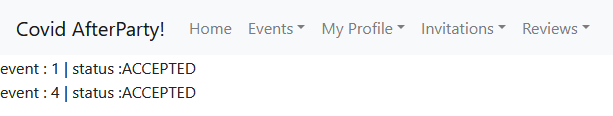
* Użytkownik musi mieć wygenerowane zaproszenie do wydarzenia
* Wydarzenie musi posiadać status COMPLETED
* Użytkownik nie może mieć już wystawionego komentarza do tej instancji wydarzenia

### Invitation Service

Moduł jest zrealizowany w sekcji Invitations z danymi opcjami menu:

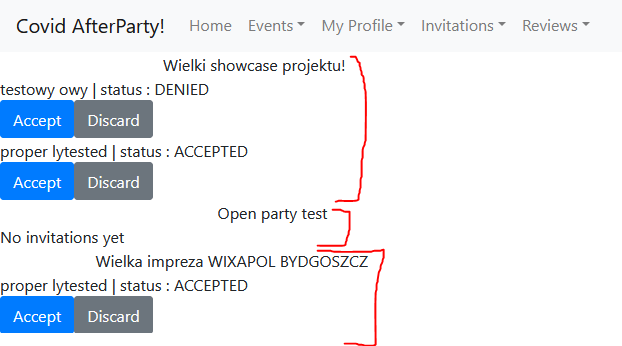


* View Invitations – pobranie listy zaproszeń użytkownika, w których jest on uczestnikiem



Użytkownik powinien widzieć status zaproszenia i przypisany event.

* Manage Event Invitations – pobranie listy zaproszeń dla organizowanych wydarzeń, w których użytkownik jest organizatorem – zarządzanie zaproszeniami dla wydarzenia



Sekcje są per wydarzenie – lista zaproszeń, gdzie organizator może akceptować lub odrzucać uczestnika. Jeżeli uczestnik wyśle zaproszenie poprzez kliknięcie „join” w module Event, zostanie utworzone, w części serwerowej, zaproszenie ze statusem PENDING\_OWNER.

Każdy użytkownik może mieć tylko jedno zaproszenie na wydarzenie, ale może uczestniczyć w wielu wydarzeniach. Użytkownik może wysłać prośbę o dołączenie do eventu (instancja zaproszenia z statusem PENDING\_OWNER), ale do momentu, gdy maksymalna liczba gości nie zostanie osiągnięta. W przypadku imprez otwartych, każde zaproszenie ma od razu status ACCEPTED.