

Kierunek: IST

ZESPOŁOWE PRZEDSIĘWZIĘCIE INFORMATYCZNE

**IwentHub – system webowy służący do zbierania i
przekazywania informacji o publicznych wydarzeniach**

Szymon Anikiej
Wojciech Gruba
Ryszard Polkowski
Agata Sito

Opiekun pracy
Dr. Inż. Krzysztof Bilewicz

Słowa kluczowe: Aplikacja Webowa, Wydarzenia, Reklama

Spis treści

DOKUMENTACJA PROJEKTOWA	3
1. Wykaz symboli, oznaczeń i akronimów:	3
2. Cel i zakres przedsięwzięcia:	3
3. Słownik pojęć:	4
4. Stan wiedzy w obszarze przedsięwzięcia:	5
5. Założenia wstępne:	9
5.1. Wymagane uprawnienia użytkowników	9
5.2. Dobór technologii	10
6. Specyfikacja i analiza wymagań na produkt programowy	11
6.1. Wymagania funkcjonalne:	11
6.2. Wymagania нефункционалне:	13
6.3. Diagram przypadków użycia:	13
6.4. Specyfikacja przypadków użycia:	14
7. Projekt produktu programowego	26
8. Implementacja:	32
9. Testy produktu programowego/Wyniki i analiza badań	33
10. Podsumowanie:	36
DOKUMENTACJA UŻYTKOWNIKA	37
1. Wprowadzenie	37
2. Instalacja produktu programowego	37
2.1. Wymagania systemowe	37
2.2. Opis procesu instalacji	37
2.3. Opis realizacji typowych zadań z podziałem na ich typy i/lub aktorów	38

DOKUMENTACJA PROJEKTOWA

1. Wykaz symboli, oznaczeń i akronimów:

W dokumentacji, aby zachować formatowanie tabel oraz diagramów wykorzystane zostało kilka skrótów, ich oznaczenia to:

G.	–	Gość
U. N.	–	Użytkownik Niezweryfikowany
U. Z.	–	Użytkownik Zweryfikowany
M.	–	Moderator
A.	–	Administrator

Dodatkowo, użyte zostały nieoczywiste skróty technologiczne, mianowicie:

JWT	–	JSON Web Token
POJO	–	Plain Old Java Objects

2. Cel i zakres przedsięwzięcia:

IwentHub jest aplikacją webową stworzoną w celu przechowywania, prezentowania i reklamowania wydarzeń publicznych.

Ideą projektu jest stworzenie systemu umożliwiającego społeczności użytkowników komunikacji o wydarzeniach różnych skali.

Aplikacja ma zatem możliwość utworzenia konta w celu zakładania własnych wydarzeń oraz deklarowania osobistych subskrypcji.

Możliwe subskrypcje organizowane przez aplikację to:

- **subskrypcja kategorii wydarzenia (np. *Koncert, Festiwal*):**
Działanie: W przypadku edycji zasubskrybowanego wydarzenia, aplikacja będzie wysyłać powiadomienia mailowe o zmianach na połączony z kontem adres mailowy.
- **subskrypcja poszczególnego wydarzenia (np. *Wrocławski Festiwal Muzyki Filmowej*):**
Działanie: Po powstaniu wydarzenia powiązanego z zasubskrybowaną kategorią, aplikacja wysyła powiadomienia mailowe na połączony z kontem adres mailowy.

Jako iż aplikacja ma służyć do reklamy i śledzenia wydarzeń, nie znaleziono potrzeby umożliwienia kupna w niej biletów na wydarzenia płatne, zamiast tego wykorzystane zostały możliwości przekierowania do stron umożliwiających zakup biletu na dane wydarzenie.

Możliwa jest integracja wydarzeń z kalendarzem zewnętrznym (np. *Google, iPhone*), w celu ułatwienia użytkownikom śledzenia interesujących ich wydarzeń.

Niezbędne w aplikacji jest wyszukiwanie i filtrowanie wydarzeń istniejących już w bazie. Robienie tego jest możliwe na wiele sposobów pozwalających użytkownikom dobrać idealne dla ich potrzeb wydarzenie.

3. Słownik pojęć:

Użytkownik – osoba korzystająca z aplikacji, na podstawie uprawnień dzieli się na:

- Gość:
 - Użytkownik niezalogowany, z najmniejszymi uprawnieniami. Oprócz rejestracji i logowania może jedynie przeglądać wydarzenia i przekazywać informacje o nich do kalendarza zewnętrznego.
- Użytkownik niezweryfikowany:
 - Użytkownik zalogowany na konto społecznościowe o podstawowych uprawnieniach. Ma dostęp do pełnej funkcjonalności swojego typu konta oferowanej przez aplikację, aczkolwiek wydarzenia dodawane przez niego wymagają aprobaty moderatora zanim zostaną udostępnione publicznie.
- Użytkownik zweryfikowany:
 - Użytkownik zalogowany na konto społecznościowe o najwyższych uprawnieniach. Są to głównie duże grupy organizujące wydarzenia (*np. Spotify, WOŚP*) oraz mała grupa lojalnych, zaufanych użytkowników. Ma dostęp do pełnej funkcjonalności swojego typu konta oferowanej przez aplikację, a wydarzenia dodawane przez niego nie wymagają aprobaty moderatora przed publicznym udostępnieniem.
- Moderator:
 - Użytkownik zalogowany na konto zarządzające o podstawowych uprawnieniach. Nie ma możliwości tworzenia wydarzeń, jedynie je weryfikuje, a następnie zatwierdza bądź odrzuca, dodawane przez użytkowników wydarzenia.
- Administrator:
 - Użytkownik zalogowany na konto zarządzające o najwyższych uprawnieniach. Nie ma możliwości tworzenia wydarzeń, jedynie je weryfikuje, a następnie zatwierdza bądź odrzuca. Ma też możliwość zmiany przywilejów poszczególnych użytkowników oraz zarządzania kategoriami wydarzeń.

Konto – zbiór danych użytkownika, w zależności od uprawnień dzieli się na:

- Społecznościowe – pozwala na tworzenie wydarzeń i korzystanie z systemu subskrypcji
- Zarządzające – pozwala na weryfikację wydarzeń oraz na wyższym stopniu uprawnień na zarządzanie użytkownikami i kategoriami

Wydarzenie – publiczne zajście (*najczęściej rozrywkowe*), reklamowane przez użytkownika w aplikacji. Definiowane jest przez następujące właściwości:

- Nazwę
- Listę kategorii
- Datę rozpoczęcia i zakończenia
- Miejsce
- Maksymalną możliwą ilość uczestników
- Kategorię wiekową
- Odpłatność
- Konieczność rezerwacji miejsca

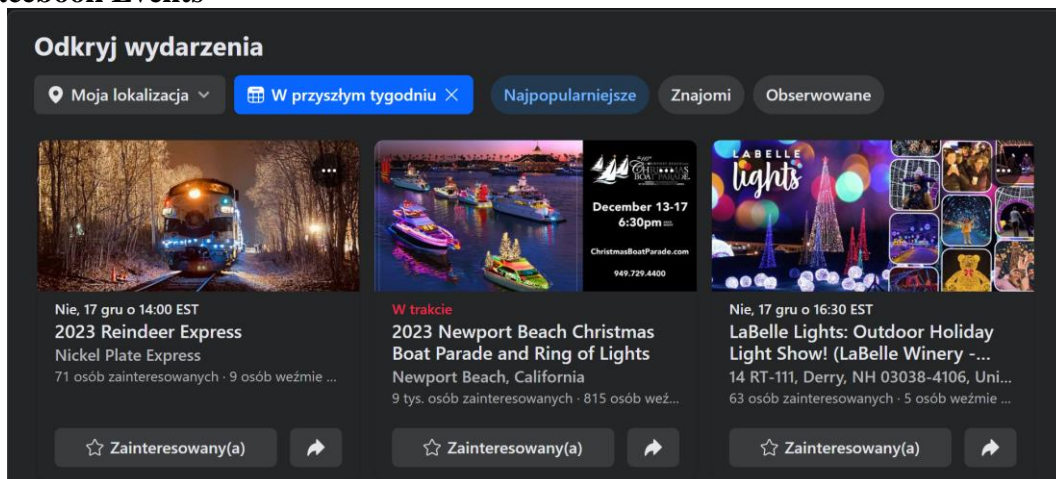
Kategoria – słowa kluczowe opisujące szeroką gamę wydarzeń. Zawierają w sobie podkategorie uściślające rodzaj wydarzenia i ułatwiające szczegółowe wyszukiwanie.

4. Stan wiedzy w obszarze przedsięwzięcia:

Aplikacja nie jest pierwszą i jedyną w swoim rodzaju, należy więc spojrzeć na jej konkurentów i zobaczyć co można wyciągnąć z ich implementacji. Analizowane w tym celu systemy to:

1. Facebook Events
2. All Events In City
3. StubHub

Facebook Events

A screenshot of the Facebook "Utwórz wydarzenie" (Create event) form. The form is titled "Utwórz wydarzenie" and has a close button (X) in the top right corner. Below the title is a "Dodaj" (Add) button. The form is filled out by "Agata Sito", the organizer, with the profile picture and name visible. The form fields include: "Nazwa wydarzenia" (Event name), "Data rozpoczęcia" (Start date) set to "14 gru 2023", "Czas rozpoczęcia" (Start time) set to "12:00", a section for "+ Data i godzina zakończenia" (End date and time), a dropdown menu for "Czy wydarzenie jest offline czy online?" (Is the event offline or online?), a dropdown menu for "Kto może to zobaczyć?" (Who can see this?), and a text area for "Jakie są szczegółowe informacje na temat tego wydarzenia?" (What are the detailed information about this event?). At the bottom of the form is a large "Utwórz wydarzenie" (Create event) button.

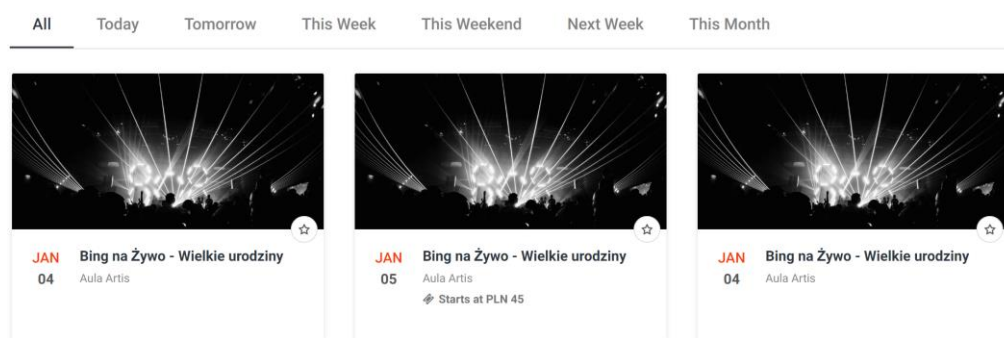
Zalety	Wady
Możliwość dodawania wydarzeń przez wszystkich	Brak moderacji treści wydarzeń
Opcja wydarzeń online	Brak głębszej kategoryzacji wydarzeń
Możliwość wyszukiwania wydarzeń	Mało wbudowanych filtrów
Możliwość obserwowania wydarzeń	Brak integracji z kalendarzem
Wygodna dostępność dla częstych użytkowników portalu Facebook	Brak komunikacji z użytkownikiem poza portalem Facebook
Niezaśmiecony interfejs przeglądu wydarzeń	Mało dynamiczne przedstawienie propozycji wydarzeń

Facebook Events jest typowo przystosowane pod model portalu Facebook – większość opcji filtrowania i ograniczeń opiera się na systemie „znajomych”. Efektywnie odcinając użytkownika od nieznajomych osób, dzielących ich zainteresowania.

All Events In City



Popular Events in Poznan



PUBLISH YOUR EVENT

Step 1

Event Name *

Enter the name of your event

Event Type *

☒ Single Event ☐ Recurring Event

Start Time *

2023-12-15

12:00 AM

+ Add End Time

Where will your event take place?

Please select an option.

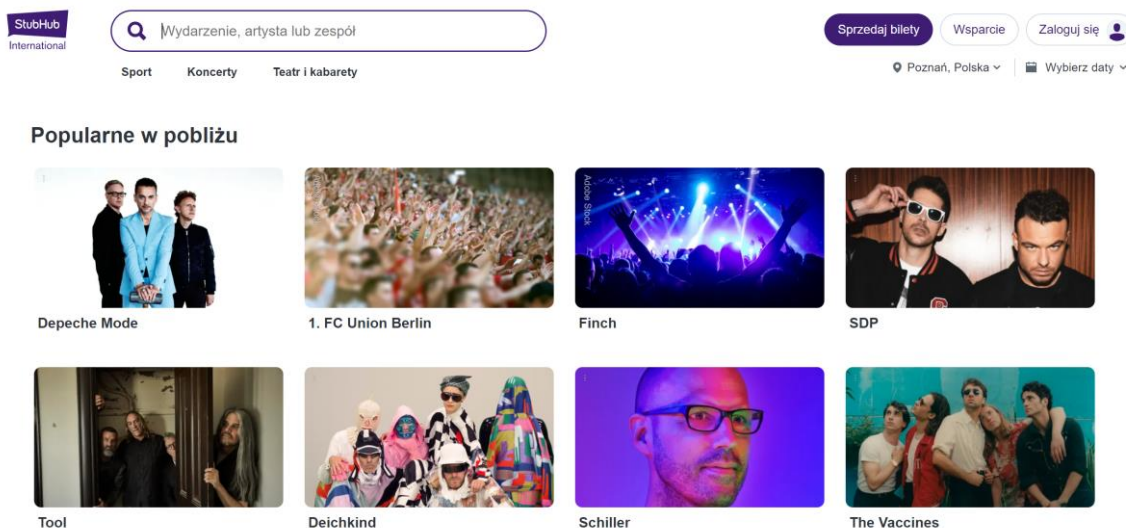
Event Description *

B **I** **U** **≡** **|** **Generate using AI**

Zalety	Wady
Możliwość subskrypcji wydarzeń i kategorii	Mało wbudowanych filtrów
Integracja z kalendarzem zewnętrznym	Spory spam w kalendarzu i skrzynce pocztowej
Możliwość dodawania wydarzeń przez wszystkich	Wyraźny nacisk na duże, płatne wydarzenia
Opcja wydarzeń online	Skupienie na wydarzeniach w pojedynczym mieście
Możliwość wyszukiwania wydarzeń	Mało czytelne wyświetlanie, dużo przewijania

All Events in City jest aplikacją stworzoną dla osób chcących zarobić na publikowanych wydarzeniach. Z graficznej perspektywy prezentuje się zimno i biznesowo.

StubHub



Zalety	Wady
Wyszukiwanie wydarzeń	Brak możliwości tworzenia wydarzeń
	Aplikacja typowo komercyjna
Rekomendacja wydarzeń	Mało dostępnych kategorii
	Nieprzyjazny interfejs

StubHub reklamuje wydarzenia w kontekście sprzedaży biletów na nie. Ze wszystkich konkurentów – przy założonych celach wypada najgorzej.

Polecane wydarzenia



Spektakl Teatralny:
Światło i Cień



Pokaz Sztucznych
ogni nad Jeziorem



Festiwal Sztuki
Ulicznej



Wystawa Sztuki
Nowoczesnej



Impreza

Aplikacja umożliwia dogłębne wyszukiwanie interesujących użytkownika wydarzeń. Przedstawia mu też propozycje wydarzeń w dynamiczny i interesujący sposób. System subskrypcji informuje użytkownika tylko o pożądanym przez niego tematach, nie spamuje jego skrzynki pocztowej.

Społeczność zbudowana jest na idei zarówno udostępnienia możliwości poznawania nieznajomych o wspólnych zainteresowaniach, jak i odnajdowania wielkich imprez. Co ważne – interfejs sprawia wrażenie ciepła i przytulności.

5. Założenia wstępne:

Wymagane uprawnienia użytkowników	G.	U. N.	U. Z.	M.	A.
Rejestracja	x				
Logowanie się	x				
Wylogowanie się		x	x	x	X
Przegląd własnych danych		x	x	x	X
Edycja własnych danych		x	x	x	X
Przeglądanie wydarzeń	x	x	x	x	X
Wyszukiwanie wydarzeń z filtrem i sortem	x	x	x	x	X
Integracja wydarzenia z kalendarzem zewnętrznym	x	x	x	x	X
Tworzenie wydarzeń ¹		x	x		
Edycja własnych wydarzeń ¹		x	x		
Usuwanie własnych wydarzeń		x	x		
Subskrypcja wydarzeń		x	x		
Subskrypcja kategorii		x	x		
Odsubskrybowanie wydarzenia		x	x		
Odsubskrybowanie kategorii		x	x		
Akceptacja wydarzeń innych użytkowników				x	X
Odrzut wydarzeń innych użytkowników				x	X
Przegląd użytkowników					X
Nadawanie praw użytkownikom					X
Odbieranie praw użytkownikom					X
Tworzenie kategorii					X
Edycja kategorii					X
Usuwanie kategorii					X
Tworzenie podkategorii					X
Edycja podkategorii					X
Usuwanie podkategorii					X

Możliwości użytkowników ograniczamy w zależności od typu konta:

- **Konta społecznościowe** – użytkownik zweryfikowany, niezwyfikowany
Pozwalają na korzystanie ze społecznych funkcji subskrypcji i tworzenia wydarzeń.
- **Konta zarządzające** – moderator, administrator
Pozwalają na zarządzanie użytkownikami i ich wkładem w społeczność oraz kategoriami.

Decyzja wprowadzenia ów podziału została podjęta w celu ograniczenia możliwości faworyzowania użytkowników o wyższych uprawnieniach oraz w celu dodatkowego zabezpieczenia przed „spamem” wydarzeń.

Np.: w przypadku gdy skutecznie zostanie zaatakowane konto moderatora, nie ma możliwości, iż utworzy on nagle proceduralnie wiele śmieciowych, lub obraźliwych wydarzeń.

¹ Dla użytkownika niezwyfikowanego spowoduje to przekazanie wydarzenia do akceptacji przez moderatora, a w przypadku użytkownika zweryfikowanego spowoduje to publikację wydarzenia użytkownikom

Dobór technologii		
Baza Danych	MongoDB	Wybrana z powodu elastyczności formatu przechowywanych dokumentów, pozwala m.in. na ewentualne przyszłe rozszerzenia zbieranych informacji (w szczególności – o wydarzeniach) bez wstecznego rozszerzania danych wartościami pustymi. Pomaga też uniknąć implementacji nieproporcjonalnie dużej ilości związków „wiele-do-wielu”, które wystąpiłyby w bazie relacyjnej.
Zabezpieczenia	BCrypt	Wybrane w celu szyfrowania przechowywanych haseł, aby utrudnić potencjalnym napastnikom uzyskanie dostępu do danych użytkowników. Wykorzystywany jest zarówno przy logowaniu jak i rejestracji.
	JWT	Wybrane z powodu lekkości, samodzielności i przezroczystości przechowywanego formatu danych. Możliwe jest też ustawienie okresu ważności, co umożliwia większą kontrolę dostępu.
Backend	Spring	Wybrany z powodu szerokiej możliwości rozwiązań i obszernej dokumentacji. Jest on modularny, pozwala na korzystanie tylko z potrzebnych programiście funkcji, przez co umożliwia szybkie i skuteczne tworzenie wysoce skalowalnych, wydajnych aplikacji. Usprawnia również konfigurację, gdyż w większości przypadków w pełni ją automatyzuje.
Frontend	Angular	Wybrany z powodu licznych gotowych komponentów zawartych w nim, pozwalających (aczkolwiek, po wprowadzeniu pewnych edycji) na bezproblemowe dobranie zgrabnie wyglądającego i wygodnego w użyciu interfejsu graficznego użytkownika. Umożliwia on też dynamiczne obsługiwanie zdarzeń, gwarantując wysoką responsywność interfejsu użytkownika na jego działania.

6. Specyfikacja i analiza wymagań na produkt programowy

Wymagania funkcjonalne:

- **Rejestracja/Logowanie:**

Goście mogą zarejestrować się w systemie lub zalogować się. Użytkownicy zalogowani mają dostęp do wszystkich należnych im (zgodnie z nadanymi prawami dostępu) funkcjonalności systemu.

- **Strona powitalna:**

Wszyscy użytkownicy mogą przeglądać stronę powitalną, aczkolwiek jej zawartość jest zależna od typu odwiedzającego ją użytkownika:

Konta społecznościowe/Gość	Konta zarządzające
<p>Najważniejszym elementem strony powitalnej jest propozycja wydarzeń, które mogą zainteresować użytkownika w trzech interaktywnych karuzelach:</p> <ul style="list-style-type: none">• Nadchodzące wydarzenia w Polsce• Nadchodzące wydarzenia w okolicy• Polecane wydarzenia <p>Każda pozycja w karuzeli jest kartą ze zdjęciem wydarzenia i jego nazwą. W momencie najechania na pozycję myszką, ujawniają się krótkie, podstawowe informacje o wydarzeniu. Kliknięcie pozycji powoduje przejście do szczegółowej strony wydarzenia.</p>	<p>Najważniejszym elementem strony powitalnej jest lista wydarzeń, które oczekują na akceptację. Prezentowane są w postaci listy. Każda pozycja jest kartą ze zdjęciem wydarzenia i jego nazwą. Dodatkowo – znajduje się na nich przycisk przenoszący do szczegółów danego wydarzenia.</p>
<p>Oprócz tego, na stronie powitalnej (<i>jak na każdej</i>) znajduje się nagłówek strony, dający dostęp do:</p> <ul style="list-style-type: none">• Strony „O nas”• Strony dodania wydarzenia (<i>Użytkownik zalogowany na koncie społecznym</i>)• Strony „Pomoc”• Wyszukiwania wydarzeń• Logowania/Rejestracji (<i>Gość</i>)• Profilu użytkownika (<i>Użytkownik zalogowany</i>)	<p>Oprócz tego, na stronie powitalnej (<i>jak na każdej</i>) znajduje się nagłówek strony, dający dostęp do:</p> <ul style="list-style-type: none">• Strony „O nas”• Strony „Pomoc”• Wyszukiwania wydarzeń• Profilu użytkownika

- **Wydarzenia:**

Wszyscy użytkownicy mogą przeglądać i wyszukiwać wydarzenia oraz korzystać z opcji integracji ich z kalendarzem zewnętrznym, jednak zarządzanie nimi jest zależne od typu użytkownika:

Gość	Konta społecznościowe	Konta zarządzające
BRAK	Dodawanie nowych wydarzeń	Akceptacja wydarzeń użytkowników niezweryfikowanych
	Edycja własnych wydarzeń	Odrzucenie wydarzeń użytkowników niezweryfikowanych
	Usuwanie własnych wydarzeń	Usuwanie wydarzeń innych użytkowników

- **Kategorie:**

Służą wszystkim użytkownikom do opisywania wydarzeń w celach łatwiejszego wyszukiwania.

Zarządzanie nimi jest dozwolone tylko administratorom.

- **Użytkownicy:**

Każdy użytkownik może zarządzać swoim kontem w następujące sposoby:

- Zmiana własnych danych
- Resetowanie hasła

Administrator może dodatkowo zarządzać prawami poszczególnych użytkowników i blokować złośliwe konta.

- **System subskrypcji:**

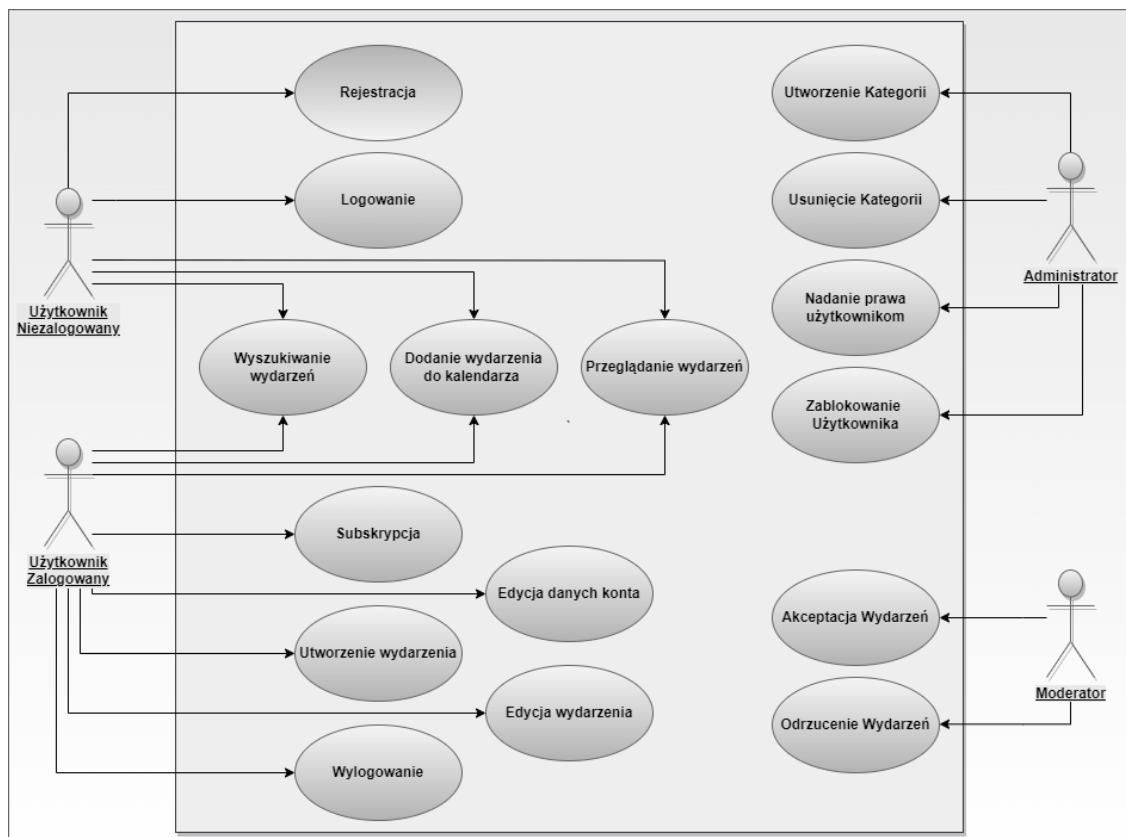
Konta społecznościowe mają dostęp do systemu subskrypcji. Istnieją dwa typy subskrypcji:

- Kategorie – powiadomienia mailowe o nowych wydarzeniach w danej kategorii
- Wydarzenia – powiadomienia mailowe o jakiegokolwiek edycji wydarzenia

Wymagania niefunkcjonalne:

- **Wydajność:**
System musi być płynny i szybki w działaniu – nawet w przypadku wysokiej ilości aktualnych użytkowników. Czas odpowiedzi na przesyłane żądania nie powinien zatem przekraczać ok. 2s.
- **Bezpieczeństwo:**
System musi przysyłać wszelkie dane użytkowników w bezpieczny, odporny na ataki sposób. Hasła wykorzystywane przez użytkowników powinny być szyfrowane przed przechowywaniem ich w bazie danych.
- **Skalowalność:**
System musi mieć możliwość łatwego rozszerzenia kategorii wydarzeń, w celu obsłużenia przyszłych nowopowstałych typów atrakcji i wydarzeń społecznych.
- **Interfejs użytkownika:**
System musi być obsługiwany przez intuicyjny interfejs graficzny, czytelny na wielu typach urządzeń i automatycznie skalujący się do wyświetlanej rozdzielczości.
- **Dostępność:**
System musi być dostępny zarówno na urządzeniach mobilnych, jak i stacjonarnych. Powinien również zapewnić pełny, wygodny użytek osobom z szczególnymi potrzebami (np.: daltonizm).

Diagram przypadków użycia:



Specyfikacja przypadków użycia:

Sekcja	Treść
Nazwa	Rejestracja
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce utworzyć konto w systemie. W tym celu wprowadza on swoje dane, system sprawdza je i w przypadku poprawności (i braku konfliktu) tworzy konto.
Aktorzy	Użytkownik niezalogowany (Gość)
Warunki wstępne	Posiadanie adresu e-mail
Warunki końcowe	Poprawność wprowadzonych danych
Rezultat	Utworzenie konta użytkownika
Scenariusz główny	<ol style="list-style-type: none">1. Użytkownik klika na przycisk „Zarejestruj”.2. Użytkownik wprowadza dane:<ul style="list-style-type: none">• Nazwę użytkownika,• Adres e-mail,• Hasło,• Powtórzenie hasła.3. System sprawdza poprawność hasła w czasie rzeczywistym.4. Użytkownik klika przycisk „Zarejestruj”.5. System sprawdza poprawność nazwy użytkownika i adresu e-mail.6. System zwraca komunikat o pomyślnej rejestracji i przenosi użytkownika na stronę logowania.
Scenariusz alternatywny	<ol style="list-style-type: none">3.A. Gdy hasło i jego powtórzenie nie spełniają warunków, system nie pozwala użytkownikowi przejść dalej.<ol style="list-style-type: none">3.A.1 Użytkownik poprawia hasło aż znikną komunikaty o jego błędzie.3.A.2 Wejście do punktu 4 scenariusza głównego.5.A. System wykrywa konflikt maila lub nazwy w bazie danych.<ol style="list-style-type: none">5.A.1 System zwraca informację, iż istnieje już konto o danej nazwie użytkownika lub adresie e-mail.5.A.2 Użytkownik wpisuje nowe wartości tych danych.5.A.3 Wejście do punktu 5 scenariusza głównego.

Sekcja	Treść
Nazwa	Logowanie
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce zalogować się na swoje konto. W tym celu wprowadza on swoje dane, system sprawdza je i w przypadku poprawności – loguje go.
Aktorzy	Użytkownik niezalogowany (Gość)
Warunki wstępne	Posiadanie konta
Warunki końcowe	Poprawność wprowadzonych danych
Rezultat	Zalogowanie na konto użytkownika
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Zaloguj”. 2. Użytkownik wprowadza dane: <ul style="list-style-type: none"> • Nazwę użytkownika lub e-mail, • Hasło. 3. Użytkownik klika przycisk „Login”. 4. System sprawdza poprawność nazwy użytkownika/adresu e-mail i wprowadzonego hasła. 5. System zwraca komunikat o pomyślnym zalogowaniu i przenosi użytkownika na stronę główną.
Scenariusz alternatywny	<ol style="list-style-type: none"> 4.A. System wykrywa niezgodność wprowadzonych danych z przechowywanymi w bazie. <ol style="list-style-type: none"> 4.A.1 System zwraca komunikat o nieprawidłowym hasle lub nazwie użytkownika. 4.A.2 Użytkownik klika w przycisk „Nie pamiętam hasła” 4.A.3 Użytkownik wprowadza adres e-mail powiązany z kontem 4.A.4 System wysyła link do resetu hasła na podany adres e-mail 4.A.5 Użytkownik odczytuje wiadomość i klika w link 4.A.6 Użytkownik wprowadza nowe hasło spełniające warunki: <ul style="list-style-type: none"> • Min. 8 znaków, • Zawiera małą literę, • Zawiera dużą literę, • Zawiera cyfrę, • Zawiera znak specjalny. 4.A.7 System zapisuje nowe hasło 4.A.8 Wejście do punktu 1 scenariusza głównego.

Sekcja	Treść
Nazwa	Wyszukiwanie wydarzeń
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce znaleźć wydarzenia spełniające dobrane przez niego kryteria. System wyświetla mu listę wydarzeń zgodnie z nimi.
Aktorzy	Użytkownik niezalogowany (Gość), Użytkownik zalogowany
Warunki wstępne	---
Warunki końcowe	Wprowadzenie kryteriów wyszukiwania
Rezultat	Wyświetlenie listy wydarzeń
Scenariusz główny	<ol style="list-style-type: none"> Użytkownik wprowadza następujące kryteria wyszukiwania (każde z nich może zostawić puste): <ul style="list-style-type: none"> Nazwę wydarzenia, Listę kategorii, Lokalizację, Datę rozpoczęcia Datę zakończenia. Użytkownik klika przycisk „Szukaj”. System wyświetla listę wydarzeń zgodną z kryteriami oraz listę dodatkowych filtrów. Użytkownik wybiera filtry z następujących: <ul style="list-style-type: none"> Potrzeba rezerwacji miejsc, Koszt, Kategoria wiekowa, Ilość miejsc. Użytkownik klika przycisk „Filtruj”. System wyświetla listę wydarzeń zgodną z podanymi kryteriami i filtrami.
Scenariusz alternatywny	<ol style="list-style-type: none"> System nie znajduje żadnych wydarzeń i wyświetla pustą stronę. System nie znajduje żadnych wydarzeń i wyświetla pustą stronę.

Sekcja	Treść
Nazwa	Przeglądanie wydarzeń
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce przeglądać szczegóły ogłoszonych wydarzeń.
Aktorzy	Użytkownik niezalogowany (Gość), Użytkownik zalogowany
Warunki wstępne	Istnienie wydarzeń w bazie danych.
Warunki końcowe	---
Rezultat	Wgląd w wydarzenia zapisane w bazie danych.
Scenariusz główny	1. Użytkownik wybiera wydarzenie z listy wyświetlonej mu przez system. 2. System wyświetla stronę szczegółów wydarzenia.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Dodanie wydarzenia do kalendarza
Priorytet	Sukces projektu: Średni Ryzyko technologiczne: Niskie
Krytyczność	Średnia
Opis	Użytkownik chce dodać dane wydarzenie do swojego kalendarza zewnętrznego.
Aktorzy	Użytkownik niezalogowany (Gość), Użytkownik zalogowany
Warunki wstępne	Istnienie wydarzenia
Warunki końcowe	Pomyślne przesłanie danych do kalendarza
Rezultat	Dodanie wydarzenia do kalendarza
Scenariusz główny	1. Użytkownik będąc na stronie wydarzenia klika przycisk „Importuj”. 2. System przenosi użytkownika na zewnętrzną stronę importu do kalendarza. 3. Użytkownik potwierdza chęć dodania. 4. Wydarzenie zostaje dodane do kalendarza.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Wylogowanie
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce wylogować się ze swojego konta.
Aktorzy	Użytkownik zalogowany
Warunki wstępne	---
Warunki końcowe	---
Rezultat	Wylogowanie użytkownika
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Wyloguj” 2. System prosi o potwierdzenie chęci wylogowania 3. Użytkownik klika przycisk „Potwierdź” 4. System wylogowuje użytkownika
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Edycja danych konta
Priorytet	Sukces projektu: Średni Ryzyko technologiczne: Średnie
Krytyczność	Średnia
Opis	Użytkownik chce zmienić swoje hasło.
Aktorzy	Użytkownik zalogowany
Warunki wstępne	---
Warunki końcowe	<p>Nowe hasło zgodne z warunkami:</p> <ul style="list-style-type: none"> • Min. 8 znaków, • Zawiera małą literę, • Zawiera dużą literę, • Zawiera cyfrę, • Zawiera znak specjalny.
Rezultat	Zmiana hasła.
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Profil”. 2. System wyświetla jego nazwę użytkownika i e-mail. 3. Użytkownik klika przycisk „Zmień hasło” 4. System wyświetla okienko zmiany hasła. 5. Użytkownik wpisuje nowe hasło. 6. System na bieżąco sprawdza hasło w odniesieniu do warunków, dodatkowo wymaga wpisania go dwa razy. 7. Użytkownik klika przycisk „Potwierdź” 8. System zwraca komunikat o poprawnej zmianie hasła i zapisuje nowe hasło w bazie danych.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Subskrypcja
Priorytet	Sukces projektu: Średni Ryzyko technologiczne: Niskie
Krytyczność	Średnia
Opis	Użytkownik chce zasubskrybować wydarzenie lub kategorię.
Aktorzy	Użytkownik zalogowany
Warunki wstępne	Uprawnienia konta – społecznościowe (U.N., U.Z.)
Warunki końcowe	---
Rezultat	Subskrypcja wydarzenia/kategorii
Scenariusz główny	<ol style="list-style-type: none"> Subskrypcja wydarzenia. <ol style="list-style-type: none"> Użytkownik, będąc na stronie szczegółów wydarzenia klika na przycisk „Obserwuj”. System zwraca komunikat o pomyślnej subskrypcji. Strona szczegółów wydarzenia zmienia się, aby umożliwić od-subskrypcję. Subskrypcja kategorii. <ol style="list-style-type: none"> Użytkownik, będąc na stronie wyświetlania swoich subskrypcji klika w przycisk „Dodaj Subskrypcję” System wyświetla okienko pozwalające na wybranie listy kategorii z dostępnych. Użytkownik wybiera interesujące go kategorie. Użytkownik klika przycisk „Potwierdź” System dodaje listę nowo wybranych subskrypcji do istniejącej. Od-subskrypcja wydarzenia. <ol style="list-style-type: none"> Użytkownik, będąc na stronie szczegółów wydarzenia klika na przycisk „Anuluj subskrypcję”. System zwraca komunikat o pomyślnym anulowaniu subskrypcji. Strona szczegółów wydarzenia zmienia się, aby umożliwić subskrypcję. Od-subskrypcja kategorii. <ol style="list-style-type: none"> Użytkownik, będąc na stronie wyświetlania swoich subskrypcji klika w przycisk kosza przy kategorii, której subskrypcję chce anulować. System prosi użytkownika o potwierdzenie anulowania subskrypcji. Użytkownik potwierdza. System anuluje subskrypcje.
Scenariusz alternatywny	<ol style="list-style-type: none"> 3.A. Alternatywny sposób anulowania subskrypcji wydarzenia. <ol style="list-style-type: none"> 3.A.1 Użytkownik, będąc na stronie wyświetlania swoich subskrypcji klika w przycisk kosza przy wydarzeniu, którego subskrypcję chce anulować. 3.A.2 System przenosi użytkownika na stronę szczegółów wydarzenia i prosi o potwierdzenie anulowania subskrypcji. 3.A.3 Użytkownik potwierdza. 3.A.4 System anuluje subskrypcje. 3.A.5 Wejście do punktu 3.3 scenariusza głównego.

Sekcja	Treść
Nazwa	Utworzenie wydarzenia
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Średnie
Krytyczność	Wysoka
Opis	Użytkownik chce ogłosić w systemie swoje wydarzenie. Tworzy je więc i przekazuje systemowi. Ten zapisuje je w bazie danych.
Aktorzy	Użytkownik zalogowany
Warunki wstępne	---
Warunki końcowe	Wypełnienie obowiązkowych pól w formularzu: <ul style="list-style-type: none"> • Nazwa wydarzenia, • Maksymalna liczba uczestników, • Czas rozpoczęcia, • Czas zakończenia, • Lokalizacja, • Opis.
Rezultat	Utworzenie wydarzenia
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Dodaj wydarzenie” będący częścią nagłówka. 2. System wyświetla stronę formularza dodania wydarzenia. 3. Użytkownik wypełnia pola formularza. 4. Użytkownik klika przycisk „Dodaj kategorii”. 5. System wyświetla listę kategorii i ich podkategorii. 6. Użytkownik wybiera pożądaną kategorię. 7. W przypadku chęci dodania większej ilości kategorii – ponowne wejście w punkt 4. W przeciwnym wypadku, wejście w punkt 8. 8. Użytkownik klika przycisk „Zapisz”. 9. System zapisuje wydarzenie, zwraca komunikat o sukcesie i przekierowuje użytkownika na stronę jego wydarzeń.
Scenariusz alternatywny	<ol style="list-style-type: none"> 1.A. Użytkownik klika na przycisk „Dodaj wydarzenie” na stronie swoich wydarzeń. 8.A. Na pola formularza nałożone są ograniczenia: <ul style="list-style-type: none"> • Data rozpoczęcia nie może być wcześniejsza niż obecna. • Data zakończenia nie może być wcześniejsza niż data rozpoczęcia System nie pozwoli na utworzenie wydarzenia do momentu poprawy tych pól.

Sekcja	Treść
Nazwa	Edycja wydarzenia
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Średnie
Krytyczność	Wysoka
Opis	Użytkownik chce poprawić ogłoszone przez siebie w systemie wydarzenie. Edytuje je więc i przekazuje systemowi. Ten zapisuje je w bazie danych.
Aktorzy	Użytkownik zalogowany
Warunki wstępne	---
Warunki końcowe	<p>Wypełnienie obowiązkowych pól w formularzu:</p> <ul style="list-style-type: none"> • Nazwa wydarzenia, • Maksymalna liczba uczestników, • Czas rozpoczęcia, • Czas zakończenia, • Lokalizacja, • Opis.
Rezultat	Edycja wydarzenia
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik, będąc na stronie swoich wydarzeń klika na przycisk edycji znajdujący się obok wydarzenia. 2. System wyświetla stronę formularza edycji wydarzenia, wypełniając jego pola istniejącymi wartościami. 3. Użytkownik edytuje pola formularza. 4. Użytkownik dodaje kategorie. <ol style="list-style-type: none"> 4.1. Użytkownik klika przycisk „Dodaj kategorie”. 4.2. System wyświetla listę kategorii i ich podkategorii. 4.3. Użytkownik wybiera pożądaną kategorię. 4.4. W przypadku chęci dodania większej ilości kategorii, ponowne wejście w punkt 4. W przeciwnym wypadku, wejście w punkt 5. 5. Użytkownik usuwa kategorie. <ol style="list-style-type: none"> 5.1. Użytkownik klika przycisk kosza obok każdej zapisanej kategorii, którą chce usunąć. 6. Użytkownik klika przycisk „Zapisz”. 7. System zapisuje wydarzenie, zwraca komunikat o sukcesie i przekierowuje użytkownika na stronę jego wydarzeń.
Scenariusz alternatywny	<p>6.A. Na pola formularza nałożone są ograniczenia:</p> <ul style="list-style-type: none"> • Data rozpoczęcia nie może być wcześniejsza niż obecna. • Data zakończenia nie może być wcześniejsza niż data rozpoczęcia <p>System nie pozwoli na zapisanie wydarzenia do momentu poprawy tych pól.</p>

Sekcja	Treść
Nazwa	Akceptacja wydarzeń
Priorytet	Sukces projektu: Niski Ryzyko technologiczne: Niskie
Krytyczność	Niska
Opis	Użytkownik chce zaakceptować sprawdzone wydarzenie
Aktorzy	Moderator
Warunki wstępne	Sprawdzenie rzetelności i formy wydarzenia (brak wulgarności, etc.)
Warunki końcowe	Zmiana statusu wydarzenia
Rezultat	Akceptacja wydarzenia
Scenariusz główny	<ol style="list-style-type: none"> 1. Będąc na stronie szczegółów wydarzenia, użytkownik klika przycisk „Zatwierdź”. 2. System prosi o potwierdzenie decyzji. 3. Użytkownik klika przycisk „Potwierdź”. 4. System wyświetla komunikat o akceptacji wydarzenia i przekierowuje użytkownika na stronę wydarzeń oczekujących na zatwierdzenie.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Odrzucenie wydarzenia
Priorytet	Sukces projektu: Niski Ryzyko technologiczne: Niskie
Krytyczność	Niska
Opis	Użytkownik chce odrzucić sprawdzone wydarzenie
Aktorzy	Moderator
Warunki wstępne	Sprawdzenie braku rzetelności lub nieprawidłowej formy wydarzenia (obecność wulgarności, etc.)
Warunki końcowe	Usunięcie wydarzenia
Rezultat	Odrzucenie wydarzenia
Scenariusz główny	<ol style="list-style-type: none"> 1. Będąc na stronie szczegółów wydarzenia, użytkownik klika przycisk „Odrzuć”. 2. System prosi o potwierdzenie decyzji. 3. Użytkownik klika przycisk „Potwierdź”. 4. System wyświetla komunikat o odrzuceniu wydarzenia i przekierowuje użytkownika na stronę wydarzeń oczekujących na zatwierdzenie.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Utworzenie kategorii
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce utworzyć nową kategorię, gdyż popularny stał się nowy typ wydarzeń
Aktorzy	Administrator
Warunki wstępne	---
Warunki końcowe	Nie istnienie kategorii o tej samej nazwie
Rezultat	Nowa kategoria
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Kategorie”. 2. System wyświetla listę istniejących kategorii. 3. W przypadku chęci dodania pod-kategorii, użytkownik klika na nazwę kategorii, do której chce ją dodać. 4. Użytkownik wpisuje nazwę nowej kategorii i klika na przycisk plusa, aby przesłać ją systemowi. 5. System weryfikuje poprawność nazwy kategorii. 6. System dodaje kategorię i odświeża widok listy.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Usunięcie kategorii
Priorytet	Sukces projektu: Wysoki Ryzyko technologiczne: Niskie
Krytyczność	Wysoka
Opis	Użytkownik chce usunąć przestarzały typ wydarzeń.
Aktorzy	Administrator
Warunki wstępne	Istnienie kategorii
Warunki końcowe	---
Rezultat	Usunięcie kategorii
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Kategorie”. 2. System wyświetla listę istniejących kategorii. 3. W przypadku chęci usunięcia pod-kategorii, użytkownik klika na nazwę kategorii, z której chce ją usunąć. 4. Użytkownik klika na przycisk minusa obok nazwy kategorii. 5. System usuwa kategorię i odświeża widok listy.
Scenariusz alternatywny	---

Sekcja	Treść
Nazwa	Nadanie praw użytkownikom
Priorytet	Sukces projektu: Średni Ryzyko technologiczne: Średnia
Krytyczność	Średnia
Opis	Użytkownik chce zmienić prawa nadane innemu użytkownikowi.
Aktorzy	Administrator
Warunki wstępne	Istnienie innego użytkownika.
Warunki końcowe	---
Rezultat	Nadanie nowych praw użytkownikowi.
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Użytkownicy” 2. System wyświetla listę użytkowników. 3. Użytkownik wybiera użytkownika z listy i klika przycisk „Sprawdź” znajdujący się obok niego. 4. Użytkownik klika na przycisk odpowiadający poziomowi uprawnień jakie chce przypisać oglądanemu użytkownikowi. 5. System prosi o potwierdzenie decyzji zmiany uprawnień. 6. Użytkownik potwierdza zmianę. 7. System zapisuje zmianę w bazie i wyświetla ją na odświeżonej stronie oglądanego użytkownika.
Scenariusz alternatywny	<p>2.A. Istnieje możliwość filtrowania listy w celach łatwiejszego znalezienia użytkownika.</p> <p>Można robić to po następujących polach:</p> <ul style="list-style-type: none"> • Obecny poziom uprawnień, • Nazwa, • Adres e-mail.

Sekcja	Treść
Nazwa	Zablokowanie użytkownika
Priorytet	Sukces projektu: Średni Ryzyko technologiczne: Średnia
Krytyczność	Średnia
Opis	Użytkownik chce zablokować szkodliwego użytkownika.
Aktorzy	Administrator
Warunki wstępne	Istnienie innego użytkownika.
Warunki końcowe	---
Rezultat	Zablokowany użytkownik traci dostęp do systemu.
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Użytkownicy” 2. System wyświetla listę użytkowników. 3. Użytkownik wybiera użytkownika z listy i klika przycisk „Sprawdź” znajdujący się obok niego. 4. Użytkownik klika na przycisk „Zablokuj”. 5. System prosi o potwierdzenie decyzji. 6. Użytkownik potwierdza zmianę. 7. System zapisuje zmianę w bazie i wyświetla ją na odświeżonej stronie oglądanego użytkownika.
Scenariusz alternatywny	<p>2.A. Istnieje możliwość filtrowania listy w celach łatwiejszego znalezienia użytkownika. Można robić to po następujących polach:</p> <ul style="list-style-type: none"> • Obecny poziom uprawnień, • Nazwa, • Adres e-mail.

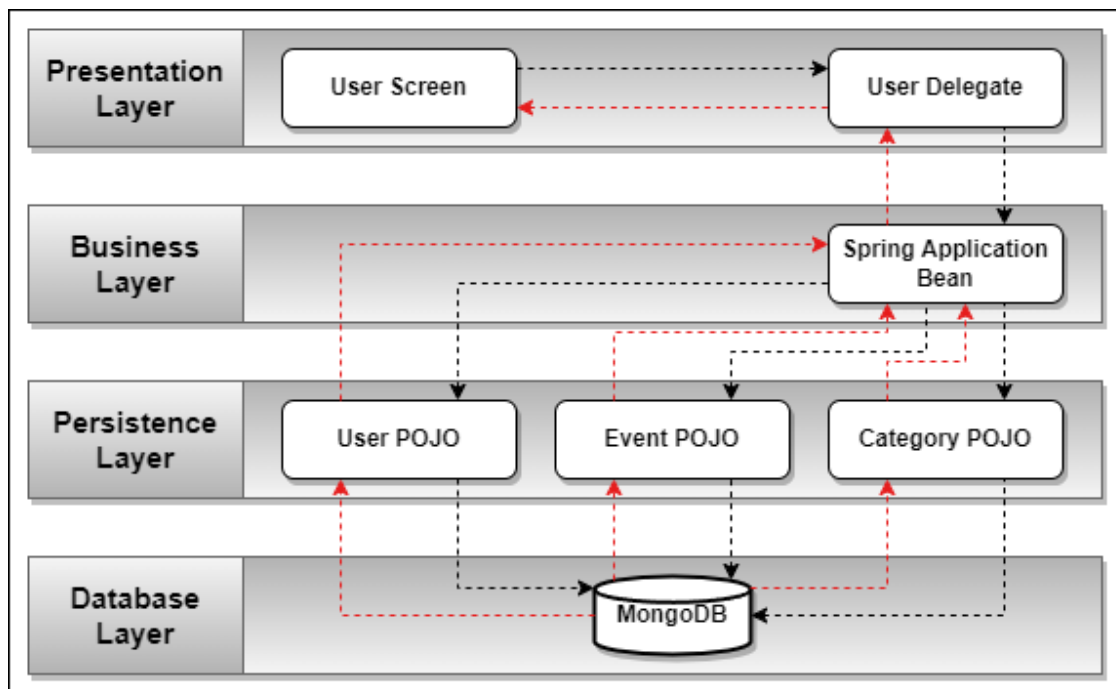
7. Projekt produktu programowego

Wybrana architektura:

W związku z faktem, iż wykorzystano do implementacji technologię Spring, **architektura warstwowa** nasuwa się samoistnie – jako, iż idealnie pasuje ona do projektu podążono tutaj za konwencją.

Zalety architektury warstwowej:	Wady architektury warstwowej:
Modularność: Umożliwia to łatwe zarządzanie, utrzymanie i rozwijanie systemu. Wprowadzenie zmian w jednym module nie wpływa na pozostałe, co przyczynia się do elastyczności i skalowalności systemu.	Zależność od innych warstw: W przypadku uszkodzenia lub przedawnienia narzędzi używanych w jednej warstwie, cały serwis nie będzie w stanie działać poprawnie. W momencie gdy warstwy są zbyt ściśle związane zmiany w jednej mogą wpływać na inne, co może skomplikować zarządzanie oraz aktualizację.
Testowanie: Skuteczne testowanie staje się bardziej osiągalne, ponieważ zmiany w jednej warstwie nie wymagają kompleksowych testów całego systemu. Dzięki temu zastosowaniu warstwy można testować osobno każdą warstwę projektu, sprawia to że jest to mniej skomplikowane, znacznie efektywniejsze oraz przyczynia się do zwiększenia jakości oprogramowania.	Wydajność i Skalowalność: W przypadku rozrostu systemu mogą pojawić się problemy z wydajnością i skalowalnością ponieważ każde zapytanie musi przejść przez wszystkie warstwy, co w przypadku bardziej rozbudowanego systemu może prowadzić do generowania przeciążeń oraz opóźnień.
Łatwość rozwoju: Dzięki łatwości rozwoju możliwe jest szybkie wdrażanie zmian, aktualizacji i dodawanie nowych funkcji do systemu. To z kolei przyczynia się do zwiększenia adaptacyjności systemu na zmieniające się potrzeby użytkowników. Jest to kluczowa zaleta tej architektury, zważywszy na ilość czasu przeznaczoną na wdrożenie działający system.	

Diagram architektury warstwowej:



Warstwy architektury aplikacji		
Nazwa warstwy:	Obsługiwane przez warstwę:	Główna technologia warstwy:
Prezentacji	GUI użytkownika oraz kontrola tego co się na nim wyświetla	Angular
Logiki biznesowej	Operacje wykonywane na pobieranych danych w celu przekazania ich dalej	Spring Boot
Utrwalania	Utrwalanie danych za pomocą opisujących je POJO	Spring Boot
Danych	Bezpośredni dostęp do bazy danych	MongoDB

Wybrana technologia frontend:

Do implementacji warstwy prezentacji wykorzystany został framework **Angular**, będący ustrukturyzowanym komponentowo szkieletem, nie bez powodu cieszącym się wielką popularnością wśród developerów tworzących aplikacje webowe.

Zdobył on to uznanie przez następujące funkcje:

- Komponentowość – oczywista pozycja na tej liście, ale niezmiernie ważna. Dzięki niej możliwe jest organizowanie aplikacji w oddzielne moduły (*lub inaczej – komponenty*), z których każdy obejmuje swoją własną logikę, widok (*szablon HTML*) oraz style. Możliwe jest zagnieżdżanie komponentów, tworzenie hierarchii, a także wielokrotne wykorzystywanie ich w różnych częściach aplikacji. Taka elastyczność znacznie ułatwia zarządzanie kodem oraz późniejsze utrzymanie aplikacji.
- TypeScript – choć język może wydawać się mniej istotny niż inne czynniki wyboru technologii, w rzeczywistości silnie na nim waży. Typowanie statyczne pomaga minimalizować problemy wynikające z typów danych, co przyczynia się do bardziej niezawodnych i łatwiejszych do utrzymania aplikacji.
- Angular CLI – nie można zlekceważyć wagi integracji technologii z narzędziami developerskimi. W tym przypadku nie ma z tym żadnych problemów, gdyż istnieje Angular CLI, czyli narzędzie za pomocą którego procesy takie jak:
 - inicjowanie projektów,
 - generowanie komponentów,
 - obsługa zależności,Są dziecinnie łatwe.
- **Routing** – Angular ma wbudowany mechanizm nawigacji między widokami, pozwalający na tworzenie aplikacji dynamicznych, które odpowiadają na działania użytkownika bez potrzeby przeładowania strony. Jego istnienie pozwala na znacznie większą interaktywność aplikacji.

Istnieje również wiele innych korzyści wynikających z używania tego frameworka, jednak w kontekście aplikacji to przede wszystkim opisane powyżej cechy miały kluczowe znaczenie dla podjętej decyzji.

Nie można jednak zapomnieć o jego wadach:

- Przede wszystkim jest on skomplikowany. Wspomniana powyżej modularność komponentów, wbudowane opcje routingu, a nawet jego bogata i rozbudowana dokumentacja mogą sprawić kłopot osobom, które dopiero zaczynają w nim pisać.
- Aplikacje w nim pisane są też mało wydajne, co dzieje się głównie z powodu skomplikowanych widoków. Jak widać więc, interaktywność interfejsów jest tu zarówno wadą i zaletą.
- Do tego, ich pliki są dość duże. W naszym przypadku nie wpłynęło to na rozwój aplikacji, aczkolwiek w przypadku ograniczonych przepustowości sieci jest to warte wzięcia pod uwagę.

Jako, iż w grupie projektowej wszyscy członkowie mieli wcześniej choć pobieżne doświadczenie z ów frameworkiem, jedyną wadą, która miała wpływ na wybór była niższa wydajność.

Padła decyzja, że ta kwestia nie jest na tyle znacząca, aby była uznawana za znaczącą niekorzyść – aplikacja nie jest w tej kwestii okropnie wymagająca.

Wybrana technologia backend:

Jak wspomniano powyżej, główną technologią, na której opiera się warstwa logiki aplikacji jest framework Spring, a dokładniej Spring Boot – narzędzie, które ułatwia i przyspiesza proces tworzenia aplikacji opartych na ów frameworku. Podobnie jak jego nadrzędna struktura, Spring Boot jest bardzo obszerną technologią. Jest jednak modularna, co pozwala programiście na korzystanie jedynie z niezbędnych funkcji, eliminując konieczność dodawania zbędnych elementów do projektu.

To właśnie ów modularność przyciąga uwagę przy poszukiwaniu technologii, przeznaczonej do implementacji. Zatrzymują ją na Springu następujące atuty:

- Automatyczna konfiguracja – ustawienia do aplikacji generowane są na podstawie użytych zależności i komponentów, co znacząco redukuje konieczność ręcznej konfiguracji przez programistę. Oczywiście istnieje możliwość nadpisania tych ustawień, ale zazwyczaj nie jest to konieczne.
- Wbudowany serwer – jako iż jego istnienie eliminuje potrzebę konfiguracji zewnętrznego serwera, aplikację można uruchomić jednym poleceniem.
- Testowalność – poprzez dostarczenie narzędzi do tworzenia testów jednostkowych i integracyjnych, Spring ułatwia znacznie proces ich pisania.

Jak w każdej technologii – w Spring Boot nie obywają się bez wad:

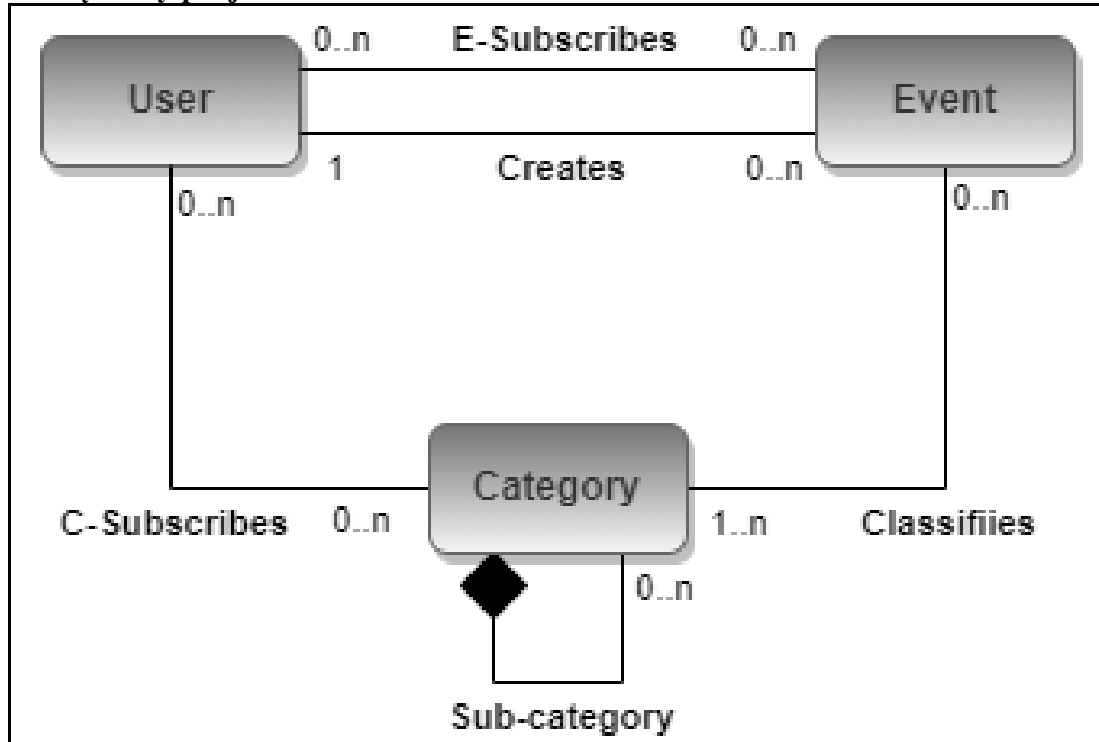
- Podobnie jak Angular, jest on niebanalny w użytku. Choć znacznie upraszcza proces stawiania aplikacji w frameworku Spring, dla niektórych dalej może się wydawać zbyt skomplikowany.
- Przystosowanie do dużych projektów – jest to coś co na naszą aplikację nie wpływa, ale należy wspomnieć iż technologia ta nie jest odpowiednia do tworzenia mniejszych, prostszych projektów. Jej funkcje są na to zbyt rozbudowane.

W przypadku aplikacji projektowej, żadna z tych wad nie ma znaczenia, a zalety wprowadzają wiele możliwości.

Wybrana baza danych:

Mniej konwencjonalny jest wybór bazy danych. Po zaprojektowaniu ERD bazy relacyjnej zauważono nieproporcjonalnie dużą ilość relacji wiele-do-wielu. Ów fakt wymusił poszukiwanie innego, lepszego rozwiązania.

Początkowy projekt ERD:



Rozwiązanie znalezione zostało w bazie dokumentowej MongoDB. Co więcej – okazało się, iż ów technologia oferuje też większą elastyczność przechowywania danych. Wszelkie rozszerzenia w wewnętrznej formie trzech podstawowych kolekcji aplikacji:

- Użytkownika,
- Wydarzenia,
- Kategorii,

nie wymagają wstecznej implementacji, gdyż bazy dokumentowe pozwalają na istnienie różnic zawartości w kolekcji – w przeciwieństwie do bazy relacyjnej, gdzie wymagana jest spójność.

Jednak jak sama nazwa wskazuje – baza dokumentowa nie obsługuje relacji pomiędzy kolekcjami. Powiązania jeden-do-wielu implementujemy poprzez wewnętrzne odwołania do wartości identyfikatora pożądanego dokumentu. Powiązania wiele-do-wielu natomiast są tworzone przez dwustronne listy identyfikatorów.

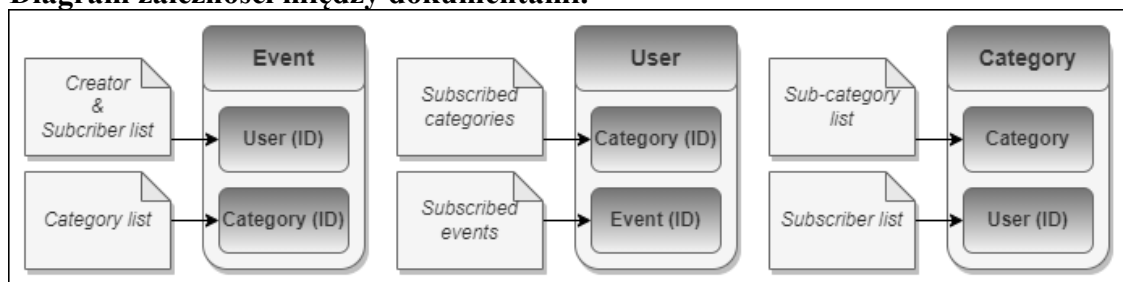
A zatem, w poszczególnych kolekcjach umieszczono następujące odwołania:

- Użytkownik:
 - Lista subskrypcji (kategorie)
 - Lista subskrypcji (wydarzenia)
- Wydarzenie:
 - Lista kategorii
 - Lista subskrybujących
 - Twórca
- Kategoria:
 - Lista subskrybujących

Można jednak zauważyć, iż tego typu implementacja naśladuje implementację bazy relacyjnej. Istotnie, po wczytaniu danych program działa na tej samej zasadzie. Można by zatem spytać czy nie jest to po prostu baza relacyjna z dodatkowymi krokami. Może i by tak było, gdyby nie fakt, iż ów interpretacja rozwiązania nie uwzględnia istnienia podkategorii.

Podkategorie są zawarte wewnątrz dokumentów ich kategorii nadrzędnych, po stronie logiki aplikacji reprezentowane są przez te same POJO co kategorie nadrzędne, ale w bazie są przechowywane za pomocą list w pełni zadeklarowanych obiektów.

Diagram zależności między dokumentami:



8. Implementacja:

Jak wspomniano powyżej – nietypowym wyborem wśród dobranych technologii w projekcie jest rezygnacja z bazy relacyjnej. Choć nie można powiedzieć, że bazy dokumentowe są czymś innowacyjnym i niespotykanym – nie mniej jednak są używane rzadziej.

Dlatego też powstał problem, który przy spojrzeniu wstecz wydaje się być bardzo oczywisty. Nastąpiło to przy integracji z innymi technologiami, w szczególności ze Springiem.

Albowiem Spring, pomimo tego, że jest frameworkiem języka Java – nie obsługuje w żaden oczywisty sposób czysto jadowej implementacji łączenia się z bazą Mongo.

Jak okazało się po fakcie – moduł Spring Data jest bardzo nieelastyczny pod względem owej implementacji.

Można by spytać - skąd to przeoczenie? Spring Data jest przecież jednym z podstawowych komponentów frameworka Spring.

Odpowiedź na to pytanie jest krótka: Przy pierwszej implementacji czytano złą dokumentację.

Dokumentacja MongoDB okazała się być bowiem lekko zdradliwa – lista sterowników znajdująca się na ich stronie jest na tyle obszerna, że uznano brak rozróżnienia implementacji przy korzystaniu ze Springa za jednoznaczne z nieistnieniem różnic.

Błąd ten był kosztowny, gdyż w pewnym momencie w pełni zatrzymał możliwość tworzenia projektu.

Na szczęście – choć była to, z punktu widzenia technologicznego, jedna z największych przeszkód jakie stanęły na drodze aplikacji: udało się ją pokonać.

Co interesujące, wystąpienie i naprawa tego problemu spowodowała równocześnie rozwiązanie innego, który tygodniami włączył się za projektem – mianowicie reprezentacji dokumentów kategorii za pomocą POJO, w szczególności przy ich edycji.

Funkcje Spring Data MongoDB są bardziej ograniczone niż funkcje czystej Javy. Zwłaszcza w przypadku opcji aktualizacji danych. Poprzednia implementacja zakładała wysoką modularność edytowanych danych, natomiast Springowe repozytoria Mongo nie pozwalają na inne opcje niż nadpisywanie całego dokumentu.

Jest to rozwiązanie gorsze ze względu na bezpieczeństwo i rzetelność danych, gdyż wprowadza ryzyko utraty znacznie większej ilości danych w przypadku wystąpienia błędu w nadpisie. Nie jest to jednak błąd o dużym prawdopodobieństwie wystąpienia i jest zminimalizowany poprzez obsługę błędów w kodzie.

Poprzednia implementacja zakładała indywidualne edycje pojedynczych pól, łącznie z podkategoriami. Było to rozwiązanie problematyczne do implementacji, gdyż wymagało utworzenia wielu bardzo szczegółowych filtrów, których wybór był w pełni zautomatyzowany poprzez własności obiektu, na którym przeprowadzane były operacje.

Okazało się, że w wykorzystanej technologii nie ma takiej opcji w żaden jasny, rzetelny sposób.

9. Testy produktu programowego/Wyniki i analiza badań

Testując aplikację głównie skupiono się na testach manualnych i wydajnościowych. Testy jednostkowe nie wносиły zbyt wielu wartościowych informacji więc zostały w porównaniu z nimi zepchnięte na drugi plan.

Testy manualne:

Każdy z projektantów oddał aplikację w ręce pięciu osób. Spośród tych dwudziestu przetestowanych:

- dwie osoby były informatykami (*koledzy z roku*),
- dziewięć potrafiło sprawnie korzystać z urządzeń elektronicznych bez znajomości programowania (*rodzeństwo/znajomi*),
- sześć miało jedynie podstawowe pojęcie o ich możliwościach (*rodzice*),
- a trzy nie znały się na nich prawie wcale (*dziadkowie*).

Co wynikło z owych testów to to, że aplikacja jest prosta w użytku. Choć niektórzy nie korzystali z pewnych funkcji (np.: nie wykorzystywali w żaden sposób filtrowania wydarzeń), podstawowe funkcje aplikacji nie sprawiły im większych problemów.

Testy pomogły jednak wykryć wiele małych błędów, jak np.: wyświetlanie się komunikatów subskrypcji wydarzenia i jej anulowania na odwrót niż być powinny, niestabilne działanie kilku funkcji, itp.

Na podstawie tych testów podjęta została decyzja na wprowadzenie kilku popraw, m.in. możliwości wyszukiwania użytkowników po imieniu.

Testy wydajnościowe:

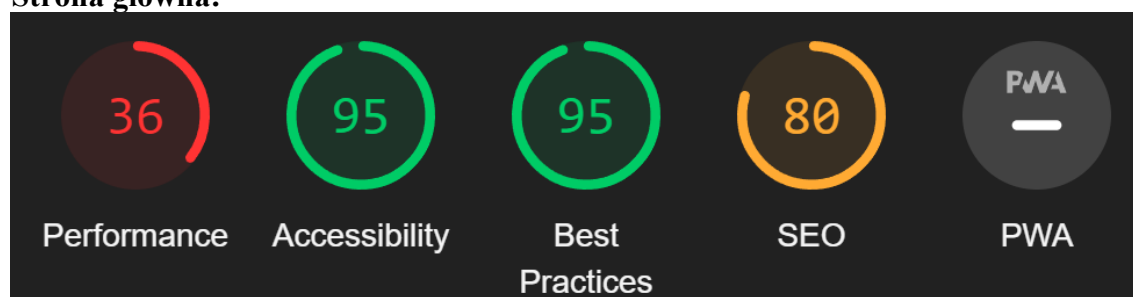
W przypadku aplikacji webowych, prawdopodobnie najważniejsze testy.

Zostały wykonane za pomocą:

- Wbudowanych opcji analizy przeglądarki Google Chrome,
- Rozszerzenia Lighthouse Google Chrome.

Wpierw, przedstawione zostały ogólne raporty audytu aplikacji odnotowane za pomocą Lighthouse.

Strona główna:



Jak widać, o ile wyniki w ramach dostępności aplikacji, najlepszych praktyk i optymalizacji dla wyszukiwarek są wysokie – wynik wydajności ma dość niską wartość.

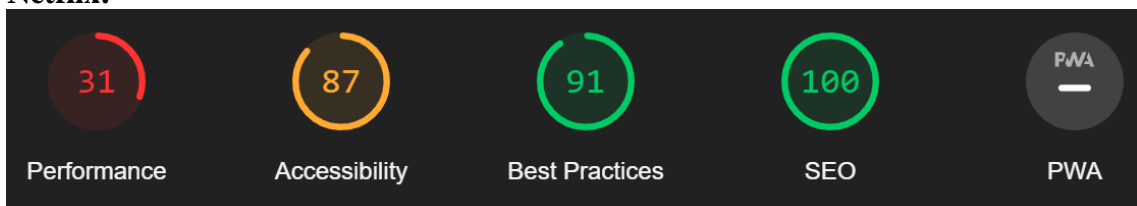
Przeanalizujemy dlaczego.

Bierze się to w dużym stopniu z implementacji dynamicznej karuzeli, za pomocą której wyświetlane są wydarzenia. Tego typu interaktywne elementy nie spisują się na testach wydajnościowych.

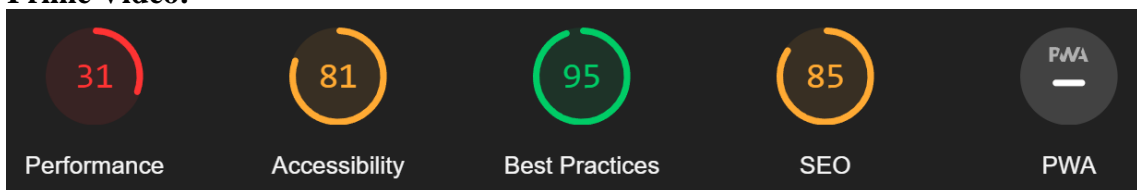
Pomimo tego – wiele portali używa je. Są one atrakcyjne dla użytkownika, prezentują wszystkie informacje jakie potrzebuje, aby zainteresować się treścią w minimalistyczny sposób, który nie przytłacza użytkownika.

Niska wydajność nie przeszkadza takim portalom jak Netflix, czy Prime Video, dlatego że prezentacja jest tu ważniejsza niż przyspieszenie ładowania o sekundę lub dwie. Co udowadniają audyty Lighthouse wykonane na ich stronach głównych.

Netflix:



Prime Video:

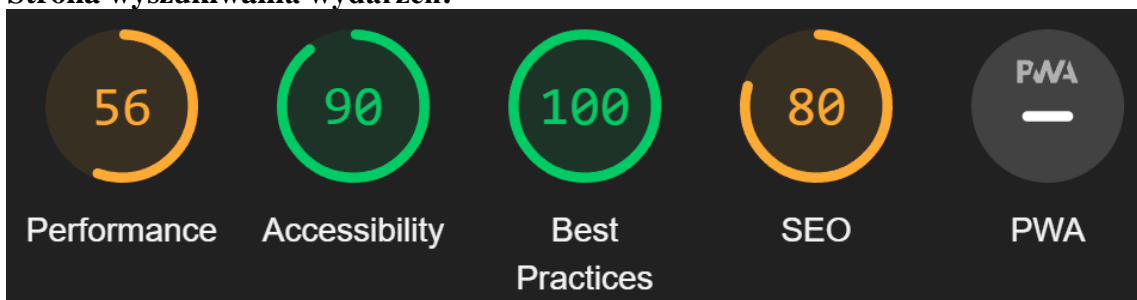


Jak widać – ich wyniki są bardzo porównywalne do aplikacji, a nawet odrobinę gorsze. Zatem strona główna i jej słabo wyglądające wyniki audytu jest w pełni akceptowalna.

Kolejna rzecz, która negatywnie wpływa na tą wartość – to dynamiczna wielkość komponentów, stąd też nawet strony, które karuzeli nie mają nie osiągają bardzo wysokiej oceny wydajności.

Jednak jak widać, znacznie wzrasta ona w momencie, gdy nie znajduje się na stronie.

Strona wyszukiwania wydarzeń:



Ale co właściwie znaczą te poszczególne kategorie?

1. Performance (Wydajność)

Opisuje szybkość i responsywność systemu. Jest też oceną rozmiaru plików, z których zawartości korzysta strona (zarówno kod jak i np. zdjęcia).

2. Accessibility (Dostępność)

Opisuje łatwość korzystania ze strony przez osoby z niesprawnościami (np.: daltonizm)

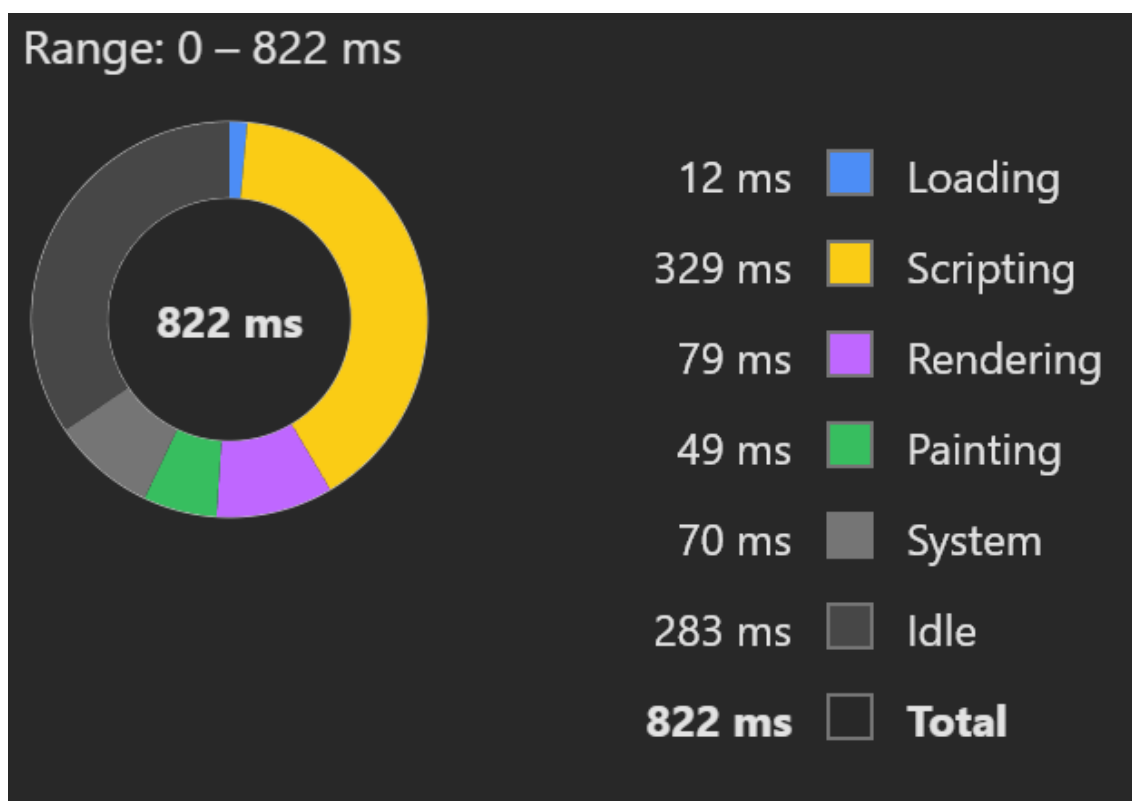
3. Best Practices (Najlepsze Praktyki)

Opisuje zgodność z aktualnymi standardami programowania.

4. SEO (Optymalizacja dla Wyszukiwarek)

Opisuje optymalizację stron w kierunku łatwości wyszukiwania ich przez wyszukiwarki internetowe.

Czyli według audytu –aplikacja ma problem z wydajnością. Weryfikacja tego faktu została wykonana poprzez test ładowania.



Jak widać, pomimo powyższego wyniku audytu strona główna ładuje się szybko. Najwięcej czasu poświęca na skrypt, który jest, m.in., odpowiedzialny za animację karuzeli co potwierdza koszt wydajnościowy elementów interaktywnych.

10. Podsumowanie:

Pomimo napotkanych problemów i opóźnień – projekt udało zrealizować się zgodnie z początkowymi założeniami.

Napotkane błędy w implementacji bazy danych spowodowane były nieznajomością technologii i jej współgrania z wykorzystywanymi frameworkami. Śmiało możemy więc stwierdzić, iż pomimo bólu jaki nam ten błąd zadał – nauczyliśmy się na jego podstawie wartościowej lekcji.

Zawsze czytaj dokumentację wszystkich wykorzystywanych technologii zanim zaczniesz cokolwiek implementować.

Pomimo przeszkód – aplikacja działa i prezentuje się z oczekiwanym efektem. A co ważniejsze, spełnia całkowicie założone wymagania.

Z punktu graficznego, aplikacja jest bardzo dynamiczna. Komponenty znajdują swoje miejsce oraz rozmiar w zależności od rozdzielczości, strona główna jest interaktywna i dynamiczna, formularze sprawdzają prawidłowość wprowadzanych danych w czasie rzeczywistym (z wyjątkiem porównywania z wartościami zawartymi w bazie danych), a wszystko to jest w otocze ładnych kolorów dynamicznie dobierających się do interfejsu w zależności od uprawnień użytkownika.

Owa dynamiczność była ogromnym wyzwaniem i jesteśmy z niej bardzo zadowoleni.

W warstwach logiki i bazy danych najwięcej wyzwań powstało przy konwersji danych z dokumentów bazy, przez POJO, na wysyłane do warstwy prezentacji obiekty JSON. Równie zdradliwa okazała się implementacja end-pointów. Zdawało się, że z każdym utworzonym odkrywaliśmy drogę do kolejnego, który będzie nam potrzebny.

Ostatecznie, pomimo potyczek na drodze i niespodziewanych przeszkód – aplikacja została wykonana poprawnie. Implementując ją, stosowaliśmy się do standardu dobrych praktyk programowania. Co najważniejsze – byliśmy w stałym kontakcie.

Każde z nas wiele się nauczyło od pozostałych członków grupy i z pewnością wynosimy z projektu wiedzę, która w przyszłości nam się przyda.

DOKUMENTACJA UŻYTKOWNIKA

1. Wprowadzenie

Ta dokumentacja jest przeznaczona na użytek osób chcących zapoznać się z systemem i jego funkcjonalnościami.

System IwentHub ma na celu umożliwienie społeczności użytkowników reklamowanie i odnajdowanie wydarzeń publicznych, które są zgodne z ich zainteresowaniami. Aplikacja umożliwia tworzenie swoich ogłoszeń, przeglądanie ogłoszeń innych, integrację z kalendarzem google i co najważniejsze – subskrypcję.

Subskrypcja jest fundamentem tworzonej w aplikacji społeczności, pozwala na powiadamianie użytkowników o wszelkich zmianach natychmiast po ich implementacji.

Oczywiście – cała ta społeczność rozwija się pod pieczęą drużyny moderatorów i administratorów.

System jest łatwo skalowalny, administrator ma pełną kontrolę nad istniejącymi w nim kategoriami wydarzeń i w każdej chwili może dodać nowe.

2. Instalacja produktu programowego

2.1. Wymagania systemowe

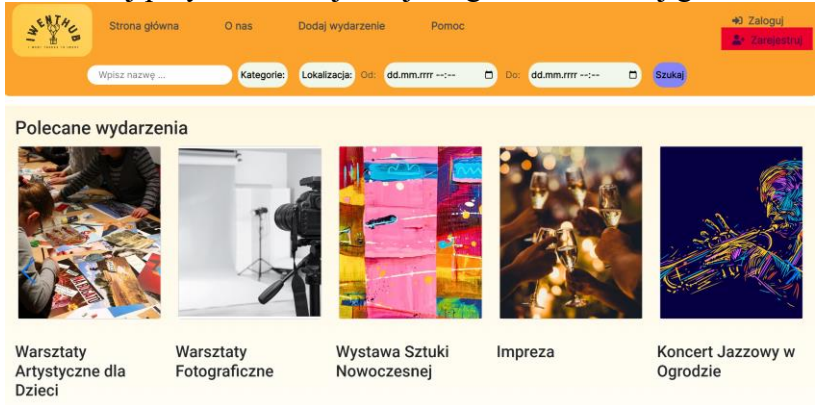
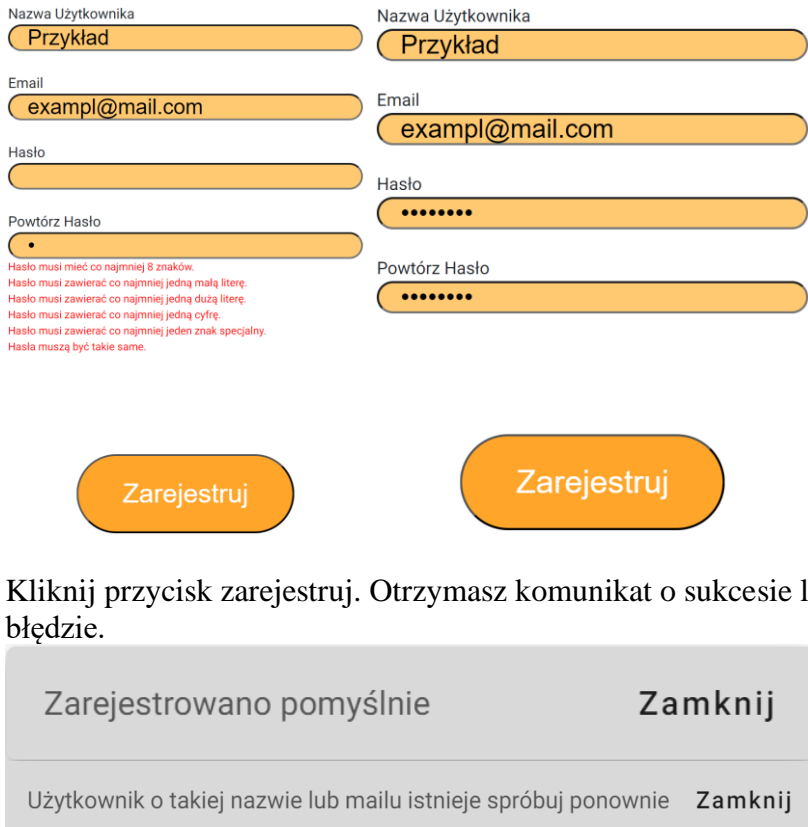
System to aplikacja webowa, dlatego do jej użytkowania nie są wymagane skomplikowane warunki. Wystarczy urządzenie z dostępem do Internetu i włączoną obsługą JavaScript w przeglądarce, np. Google Chrome, Opera lub Mozilla Firefox.


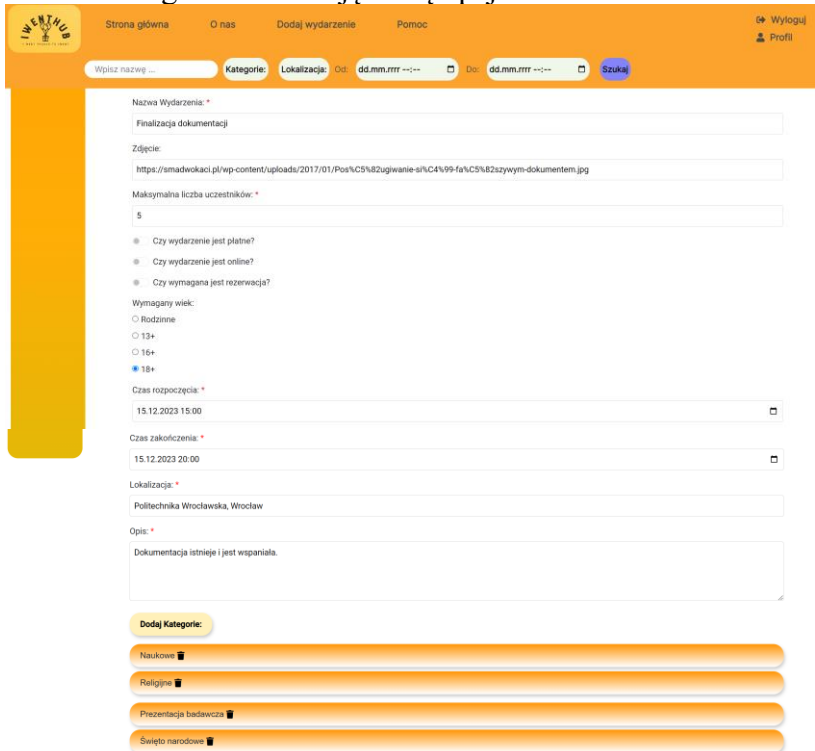
Nie polecamy korzystania z przeglądarek natywnych Microsoft, gdyż nie będą obsługiwać pewnych funkcjonalności.

2.2. Opis procesu instalacji

System nie wymaga lokalnej instalacji, wystarczy wejść na stronę.

2.3. Opis realizacji typowych zadań z podziałem na ich typy i/lub aktorów

Aktor:	Użytkownik niezalogowany (Gość)
Zadanie:	Rejestracja
Kroki:	<p>1. Zlokalizuj przycisk „Zarejestruj” nagłówka i kliknij go.</p>  <p>2. Wypełnij dane na stronie rejestracji, hasło musi spełniać widoczne na zrzucie ekranu wymagania, w momencie spełnienia wymagania – warunek znika.</p>  <p>3. Kliknij przycisk zarejestruj. Otrzymasz komunikat o sukcesie lub błędzie.</p>

Aktor:	Użytkownik nie/zweryfikowany
Zadanie:	Utworzenie wydarzenia
Kroki:	<p>1. Zlokalizuj przycisk „Dodaj wydarzenie” nagłówka i kliknij go.</p>  <p>2. Wypełnij formularz wydarzenia, pamiętając o ograniczeniach:</p> <ul style="list-style-type: none"> • Data rozpoczęcia nie może mieć miejsca przed dzisiejszą • Formularz ma wymagane pola tekstowe: <ul style="list-style-type: none"> ○ Nazwę, ○ Maksymalną liczbę uczestników ○ Czas rozpoczęcia i zakończenia, ○ Lokalizację, ○ Opis • Formularz wymusza wybór z opcji również: <ul style="list-style-type: none"> ○ Kategorii wiekowej, ○ Płatności wydarzenia, ○ Potrzeby rezerwacji, ○ Wydarzenia online. • Kategorie i link zdjęcia są opcjonalne. 

	<p>3. Kliknij przycisk „Zapisz” znajdujący się na samym dole formularza. Otrzymasz komunikat o sukcesie dodania wydarzenia.</p> <div style="background-color: #f0f0f0; padding: 10px; display: flex; justify-content: space-between; align-items: center;"> Wydarzenie zostało dodane Zamknij </div>
--	--

Aktor:	Użytkownik nie/zweryfikowany
Zadanie:	Subskrypcja kategorii
Kroki:	<p>1. Zlokalizuj przycisk „Profil” nagłówka i kliknij go.</p>  <p>2. Zlokalizuj przycisk „Subskrypcje” panela bocznego i kliknij go.</p>  <p>3. Zlokalizuj przycisk „Dodaj subskrybcję” znajdujący się na dole strony i kliknij go.</p> 

4. Kliknij przycisk „Dodaj Kategorie” i wybierz wszystkie interesujące cię pozycje z dostępnej listy.

Dodaj Kategorie:

Wybierz kategorie jakie chcesz dodać:

Dodaj Kategorie:

Anuluj **Potwierdź**

Dodaj Kategorie:


Sesja RPG	Pathfinder 1E
Festiwal	dnd 5e
Wydarzenie muzealne	Exalted Essence
Konwent	


5. Po dodaniu kategorii, kliknij przycisk „Potwierdź”.


Dodaj Kategorie:

Wybierz kategorie jakie chcesz dodać:

Dodaj Kategorie:

Sesja RPG 

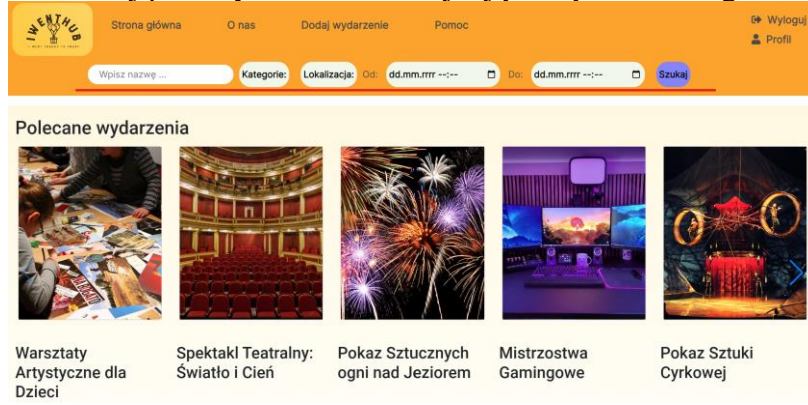
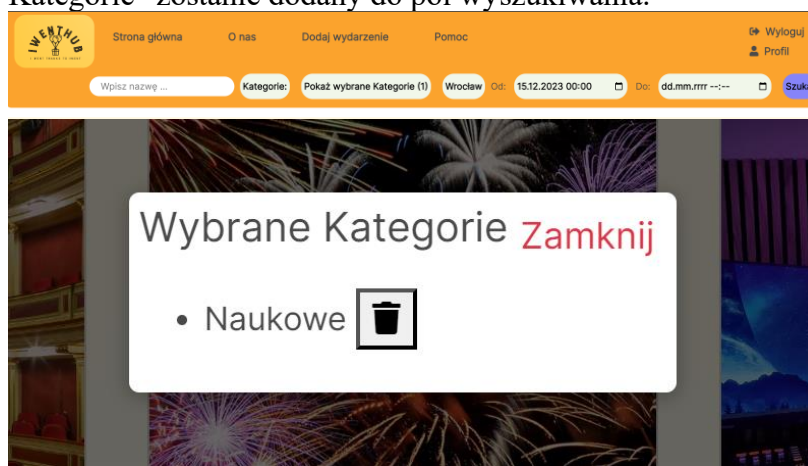
Koncert 

Rockowy 

Anuluj **Potwierdź**

6. Lista twoich subskrypcji zostanie zaktualizowana po odświeżeniu.



Aktor:	Użytkownik nie/zweryfikowany
Zadanie:	Wyszukanie interesującego wydarzenia
Kroki:	<p>1. Zlokalizuj pola wyszukiwania znajdujące się na dole nagłówka.</p>  <p>2. Wypełnij interesujące cię pola i kliknij przycisk „Szukaj”. Po dodaniu kategorii do wyszukiwania, przycisk „Pokaż wybrane Kategorie” zostanie dodany do pól wyszukiwania.</p> 

3. W celu większego zawężenia swojego wyboru zlokalizuj panel boczny zawierający listę filtrów.

The screenshot shows the INVENTUB website interface. At the top is a navigation bar with links: Strona główna, O nas, Dodaj wydarzenie, and Pomoc. On the right of the navigation bar are links for Wyloguj and Profil. Below the navigation bar is a search bar with the placeholder 'Wpisz nazwę ...'. To the right of the search bar are buttons for 'Kategorie', 'Pokaż wybrane Kategorie (1)', and 'Wrocław'. Further right are date and time filters: 'Od: 15.12.2023 00:00' and 'Do: dd.mm.rrrr --:--', followed by a 'Szukaj' button. On the left side, there is a 'Filtry' panel. It contains several sections: 'Wydarzenia już zakończone', 'Rezerwacja miejsc' (with sub-options: Rezerwacja Wymagana, Rezerwacja Niewymagana, Obojętnie), 'Koszt' (with sub-options: Darmowe, Płatne, Obojętnie), 'Wymagany wiek' (with sub-options: Rodzinne, 13+, 16+, 18+), and 'Ilość miejsc od' (with input fields for '0' and '1000', and a 'do' label). At the bottom of the filter panel are buttons for 'Filtruj' and 'Resetuj filtry'. On the right side of the page, there are two event cards. The first card is titled 'Prezentacja Aplikacji' and includes a description: 'Zapraszamy na prezentację naszej aplikacji na ZPI Obecność dobrowolna Prezentacja będzie trwała około 1h, pytania dozwolone. Oceniający Doktor Billewicz. Mamy nadzieję na 5.0'. The second card is titled 'Finalizacja dokumentacji' and includes the text: 'Dokumentacja istnieje i jest wspaniała.' Both cards have a 'Pokaż' button.

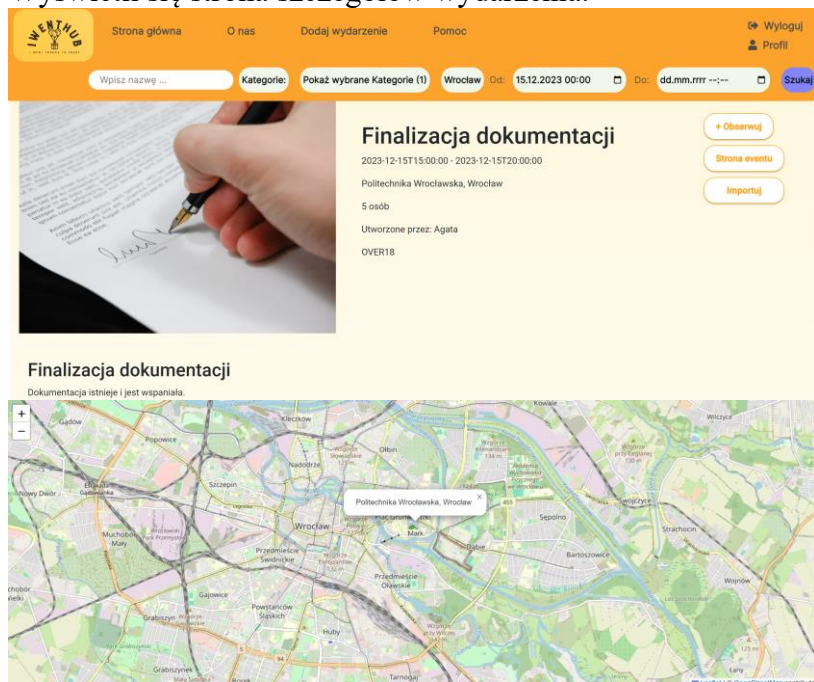
4. Wybierz filtry, które cię interesują i kliknij przycisk „Filtruj”.


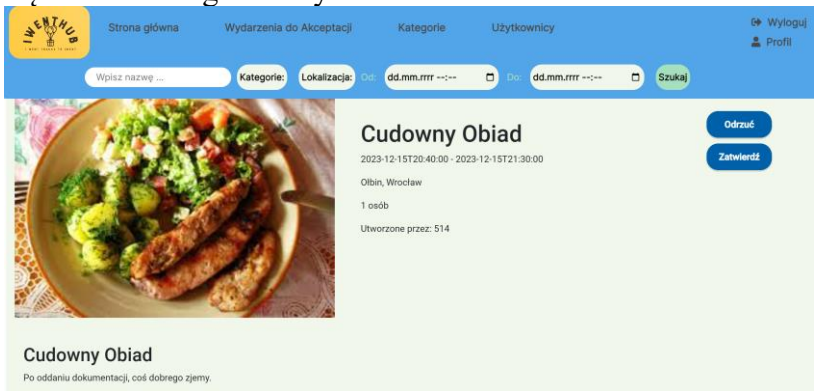
This screenshot is identical to the previous one, but with the filter panel updated. In the 'Rezerwacja miejsc' section, 'Rezerwacja Wymagana' is now selected. In the 'Koszt' section, 'Darmowe' is selected. In the 'Wymagany wiek' section, '18+' is selected. The 'Ilość miejsc od' input field now contains the value '5'. The 'Filtruj' button is still visible at the bottom of the filter panel.

5. Wyświetli się lista zawężona zgodnie z wybranymi fitrami. Wybierz interesujące cię wydarzenie z tej listy i kliknij przycisk „Pokaż”.

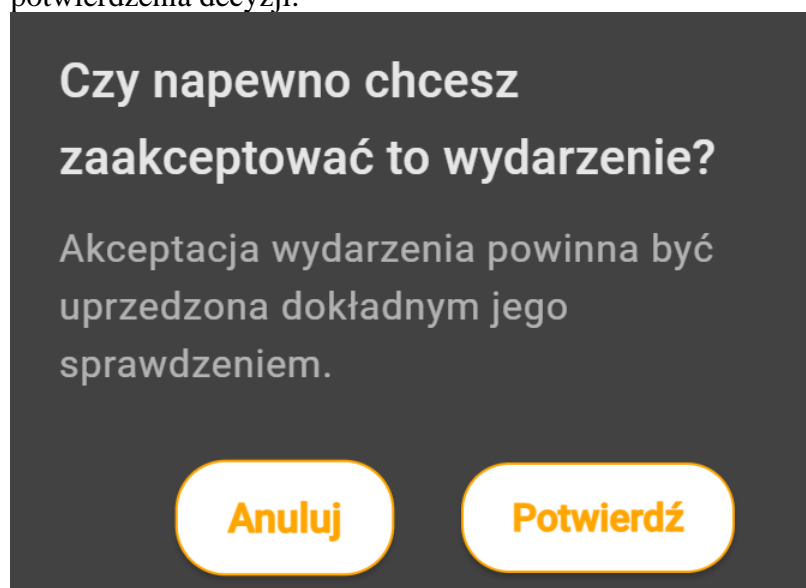
This screenshot shows the result of applying the filters. The 'Prezentacja Aplikacji' card is no longer visible. The 'Finalizacja dokumentacji' card remains, with the same description: 'Dokumentacja istnieje i jest wspaniała.' and a 'Pokaż' button. The filter panel on the left remains the same as in the previous screenshot, with 'Rezerwacja Wymagana', 'Darmowe', '18+', and '5' selected.

6. Wyświetli się strona szczegółów wydarzenia.

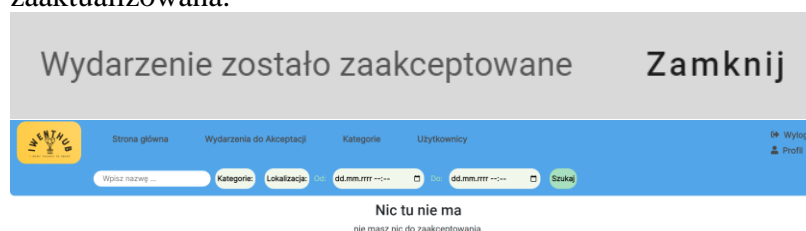



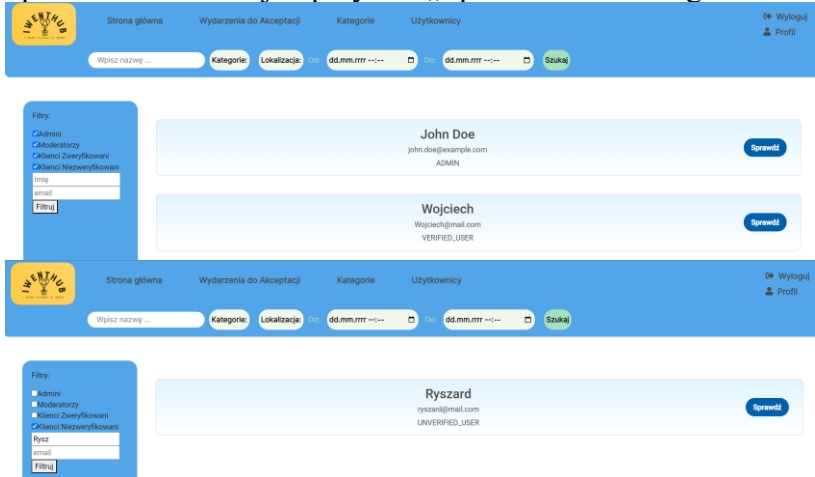
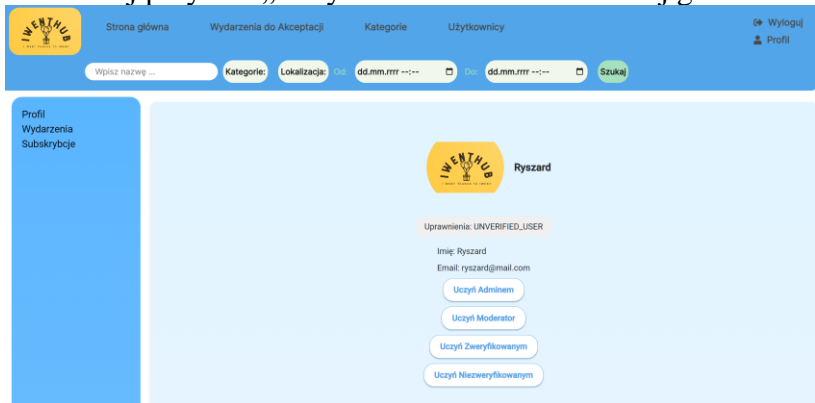
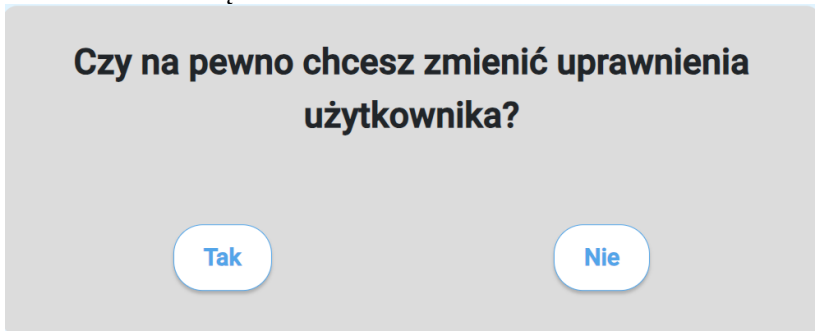
Aktor:	Moderator
Zadanie:	Sprawdzenie wydarzenia
Kroki:	<p>1. Zlokalizuj przycisk „Wydarzenia do akceptacji” nagłówka i kliknij go. Wyświetli się lista niezaakceptowanych wydarzeń.</p>  <p>2. Wybierz wydarzenie z listy i kliknij przycisk „Sprawdź”. Wyświetli się strona szczegółów wydarzenia.</p> 

3. Zweryfikuj treść wydarzenia i zgodnie z decyzją – kliknij na odpowiednik przycisk: „Odrzuć lub „Zatwierdź”. W celach dokumentacji – wydarzenie akceptujemy. Wyświetli się prośba potwierdzenia decyzji.



4. Wydarzenie zostanie zaakceptowane, wyświetlone zostanie potwierdzenie, a lista wydarzeń do akceptacji zostanie zaaktualizowana.



Aktor:	Administrator
Zadanie:	Przyznanie praw nowemu moderatorowi
Kroki:	<ol style="list-style-type: none"> <p>Zlokalizuj przycisk „Użytkownicy” nagłówka i kliknij go.</p>  <p>Korzystając z filtrów znajdź użytkownika, któremu chcesz nadać uprawnienia i kliknij w przycisk „Sprawdź” obok niego.</p>  <p>Zlokalizuj przycisk „Uczyń moderatorem” i kliknij go.</p>  <p>Potwierdź zmianę.</p> 

5. W przypadku sukcesu, na stronie użytkownika informacja o jego uprawnieniach zaktualizuje się.

