Analiza systemu powiadomień on-line o eventach realizowanych w mieście

Projekt zespołowy

Filip Kamiński, Kamil Wielgosz, Remigiusz Frankiewicz

Z712

Eventura.pl

1. **Przedstawienie koncepcji systemu**

Na rynku funkcjonuje wiele portali, które zazwyczaj ograniczają się do filtrowania i informowania użytkowników o specyficznym typie wydarzeń (np. Tylko koncerty, tylko wydarzenia teatralne etc.)

Założeniem projektu jest utworzenie systemu/aplikacji internetowej informującej użytkowników o wydarzeniach społecznych i kulturalnych w ich okolicy.

Nowo tworzony system ma być z założenia agregatorem wydarzeń ze wszystkich możliwych kategorii. Treści, które będą przedstawiane na stronie będą tworzone za pomocą dwóch metod:

* Wpisy generowane od podstaw
* Wpisy pobierane bezpośrednio od organizatorów wydarzeń za pomocą API i umieszczane bezpośrednio w systemie

Serwis eventura.pl umożliwiać będzie wyszukiwanie i filtrowanie wydarzeń w sposób prosty i przyjazny użytkownikowi na podstawie:

* Lokalizacji
* Kategorii
* Daty
* Pola wyszukiwania (słowa kluczowe)

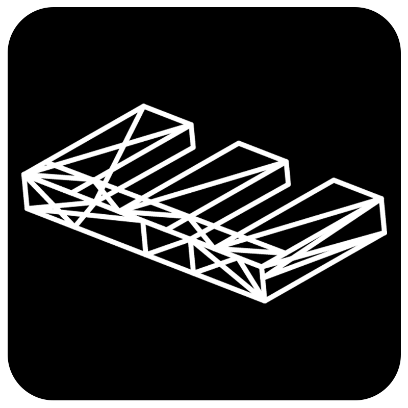
Zaimplementowane mechanizmy będą miały na celu skrócić czas wyszukiwania co skutkuje szybszym dostępem do informacji i jednocześnie przekłada się na pozytywny odbiór strony i dotarcie do jak największej liczby użytkowników (bez względu na poziom zaawansowania w obyciu z systemami internetowymi).

Serwis dodatkowo umożliwiać będzie subskrypcję interesujących użytkowników nowych wydarzeń za pomocą newsletteru oraz powiadomień push w przeglądarce internetowej - które w ostatnim czasie są bardzo popularnym sposobem na zapewnienie sobie większej liczby odwiedzin strony niż w przypadkach ,,klasycznego” marketingu internetowego.

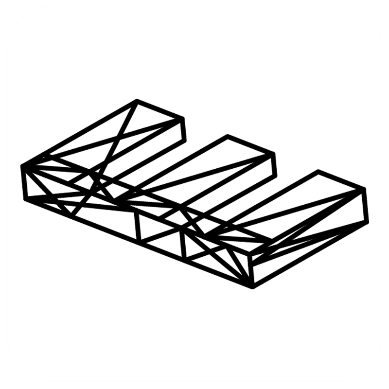
Podczas projektowania zostanie przemyślana kwestia ewentualnego przyszłego rozwoju serwisu.

Serwis pod kątem designu będzie utrzymany w motywie minimalistycznym. Formy, interfejsy oraz grafiki zawarte na stronie zostaną dobrane pod kątem uniwersalności i czytelności dla np. Różnych grup wiekowych.

**Wstępna identyfikacja wizualna serwisu**



*Sygnet wersja kolorystyczna nr 1*



*Sygnet wersja kolorystyczna nr 2*



*Wariacja kolorystyczna sygnetu*



*Typografia wersja kolorystyczna nr 1*



*Typografia wersja kolorystyczna nr 2*



*Wariacja kolorystyczna typografii*

1. **Model architektury systemu**



Powyżej został przedstawiony diagram architektury systemu.

Klient z wykorzystaniem sieci web ma nieograniczona możliwość łączenia się oraz korzystania z systemu internetowego. Serwer aplikacji złożony jest z dwóch głównych części: frontend i backend. Rozdzielenie warstwy prezentacji od logiki biznesowej oraz magazynu danych pomaga zachować wyraźny podział odpowiedzialności każdej z tych warstw. Sprzyja to także bezpieczeństwu systemu, ponieważ walidację możemy przeprowadzać na kilku etapach – wprowadzania danych, przetwarzaniu danych oraz zapisie danych. Tym samym zwiększamy bezpieczeństwo zachowania integralności danych. System zapisuje wszystkie niezbędne informacje na temat wydarzeń w bazie danych SQL.

1. **Specyfikacja funkcjonalna**

* Główna idea aplikacji:

Strona/aplikacja internetowa jako agregator informacji o imprezach i wydarzeniach kulturalnych w mieście.

* Cel biznesowy:  
  Dotarcie do jak największej liczby osób z wydarzeniami kulturalnymi w celu poszerzeniu wiedzy na temat różnego rodzaju rozrywki. Drugim celem jest ułatwienie użytkownikom odnalezienie dopasowanych do ich potrzeb i zainteresowań eventów zarówno masowych jak i kameralnych.
* Grupa docelowa aplikacji:

Osoby w każdym przedziale wiekowym, aktywne kulturowo

* Czynności oraz akcje w ramach aplikacji:

-wyszukiwanie po stringu/słowach kluczowych (np. nazwisko, nazwa itp..)

-wyszukiwanie po regionie (np. po miastach, województwach)

-wyszukiwanie po kategorii (koncerty, spektakle itp..)

-wyszukiwanie po dacie (zakres daty od-do)

-wpisy w formie listy lub kafelków z eventami na głównej stronie i podstronie po wyszukaniu

-newsletter na podstawie subskrypcji np. Regionu/miasta

-powiadomienia push w przeglądarce internetowej

* Ograniczenia we wstępnej fazie projektowania i implementacji

-brak możliwości logowania

-brak możliwości zakupu wejściówek/biletów na wydarzenia

* Integracja z innymi systemami:

-integracja z bazą danych

-API zewnętrznych dostawców za pomocą którego istnieje możliwość szybkiego pozyskiwania informacji na temat wydarzeń

* Platformy/kompatybilność:

-responsywna wersja desktop

-responsywna wersja mobile

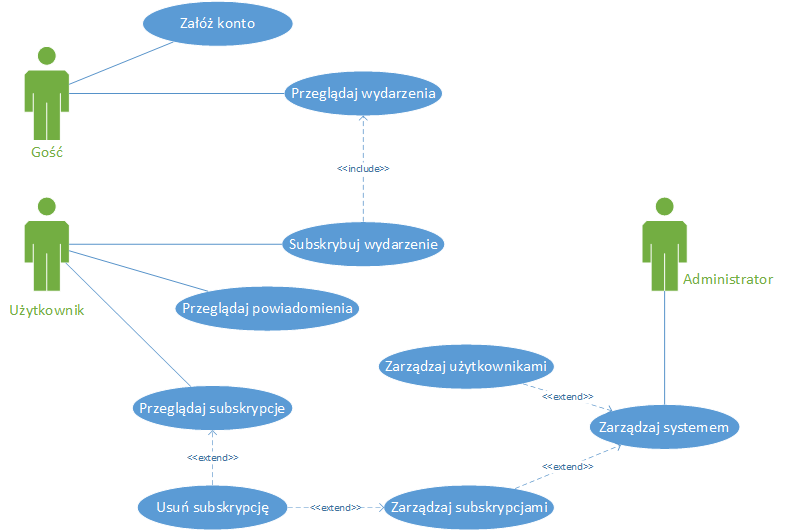
1. **Diagramy UML**

Diagram hierarchii funkcji:



Powyższy diagram przedstawia zarys dostępnych funkcji platformy. Będą dostępne 3 główne założenia biznesowe – prezentacja listy eventów, przeszukanie bazy eventów pod kątem określonych parametrów oraz powiadomienia o zbliżających się wydarzeniach. Na lista wydarzeń będzie zasilana ze źródła danych oraz będzie ona wyświetlana na stronie głównej. Wyszukiwarka będzie posiadała kilka zdefiniowanych filtrów. Z kolei subskrypcje użytkownik będzie mógł dodać, anulować oraz otrzymać zasubskrybowaną notyfikację.

Diagram przypadków użycia:



Aktorzy:

* Gość – klient serwisu bez zarejestrowanego konta. Ma tylko możliwość przeglądania wydarzeń oraz założenia nowego konta.
* Użytkownik – klient zalogowany do serwisu. Względem gościa może dodatkowo przeglądać aktualne subskrypcje, usuwać je jak i subskrybować nowe wydarzenia. Na podstawie aktualnych subskrypcji ma możliwość przeglądać powiadomienia.
* Administrator – użytkownik systemu zarządzający serwisem. Ma możliwość zarządzania kontami zarejestrowanych klientów oraz ich subskrypcjami.

Diagram przepływu danych:



Diagram przepływu danych pokazuje ogólny zarys systemu. Funkcjonalność listy wydarzeń będzie pobierała dane z magazynów danych, które będą zasilane przez zewnętrzne źródło lub manualnie. Użytkownik może wyświetlić tę listę oraz przeszukać. Dodatkowo dla każdego eventu będzie mógł dodać, anulować oraz otrzymywać powiadomienia, które będą składowane w oddzielnym magazynie danych.

1. **Model danych**



Powyżej został przedstawiony podstawowy diagram związków encji, które będą wykorzystywane w systemie. Główną częścią jest encja events – zawiera ona szczegóły dotyczące każdego wydarzenia. Łączy się ona z encją słownikową event\_category, w której przetrzymywane będą kategorie wydarzeń. W encji venues znajdują się miejsca w których odbywają się wydarzenia. Każdy event może zostać zasubskrybowany i każda subskrypcja jest składowana w encji subscriptions. W kolejnej encji słownikowej - subscription\_type – przechowywane są rodzaje subskrypcji np. push lub mail. Informacje kto dokonał subskrypcji są przechowywane w encji customers z podstawowymi danymi użytkownika. Dodatkowo istnieje encja do celów audytowych – notifications do składowania informacji o wysłanych notyfikacjach.