

Dokumentacja techniczna projektu „BiedaFilmy”

Autorzy:

Michał Jurgo

Marcin Lulkiewicz

Opis projektu

Celem projektu jest stworzenie aplikacji do obsługi bazy danych filmów. Jednocześnie ma ona udostępniać użytkownikom możliwość tworzenia własnych list filmów, oceniania ich oraz komentowania.

Stworzony program posiada bazę filmów, do której dostęp – przez interfejs przeglądarkowy – ma odwiedzający. Każdy film opisany jest wybranymi parametrami, takimi jak m.in. tytuł, rok produkcji czy obsada. Na obsadę składają się aktorzy, których dane również przechowywane są w bazie.

| Lista filmów | | | | |
|------------------|---------------------------|----------------|------------|-------------|
| Tytuł | Opis | Długość (min.) | Premiera | Gatunek |
| Ogniem i mieczem | Ale żeśmy Bohuna usiekli. | 180 | 11.11.1918 | Historyczny |

Ogniem i mieczem

Movie

| | |
|-----------------------|---------------------------|
| Tytuł | Ogniem i mieczem |
| Opis | Ale żeśmy Bohuna usiekli. |
| Długość (min.) | 180 |
| Premiera | 11.11.1918 |
| Gatunek | Historyczny |

Obsada

Michał Żebrowski jako Jan Skrzetuski

Krzysztof Kowalewski jako Jan Onufry Zagłoba

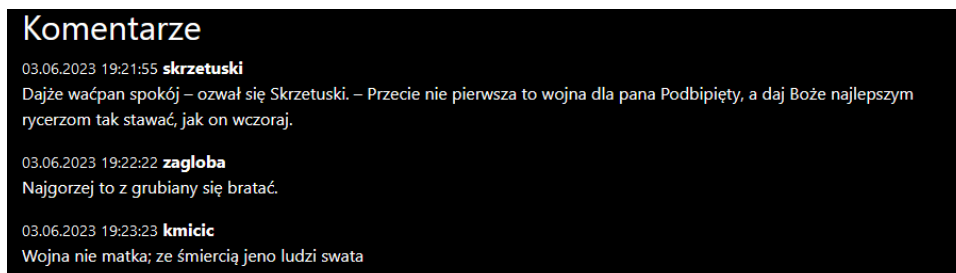
Wiktor Zborowski jako Longinus Podbięta

Na „karcie filmu” zobaczyć można również średnią jego ocen zostawionych przez użytkowników.

[Odsun komentarz](#)

Średnia ocena: 8,5

Każdy film posiada również sekcję komentarzy, w której użytkownicy wymieniają się opiniami i prowadzą niezwykle interesujące dyskusje.



Aplikacja umożliwia założenie konta w serwisie. Użytkownik posiada takie pola jak: e-mail, username oraz hasło.

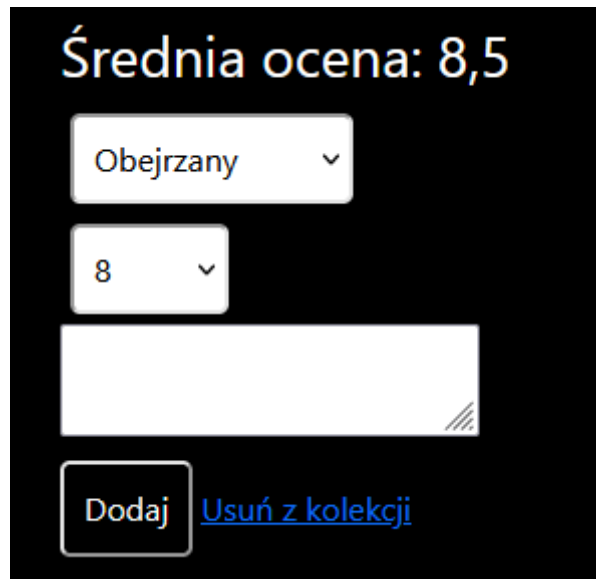
Użytkownicy mają przypisane różne role, np. „użytkownik” (rola podstawowa, może oceniać i komentować filmy, dodawać je do własnej listy), „redaktor” czy „administrator” (ma dostęp do wszystkich operacji, np. dodawania nowych filmów do bazy).

Użytkownicy rozróżniani są za pomocą adresu e-mail oraz nickname’u. Drugi z nich (razem z hasłem) służy do logowania oraz wyświetlany jest np. przy zostawionym komentarzu.

Zalogowany użytkownik otrzymuje dodatkową opcję (np. poprzez umieszczenie odpowiedniego elementu w menu) wyświetlenia własnej listy filmów. Lista ta zawiera wszystkie pozycje, które zostały przez użytkownika ocenione bądź oznaczone jako „do obejrzenia”.



Na karcie danego filmu zalogowany użytkownik ma dodatkową opcję dodania oceny, która niewidoczna jest dla niezalogowanych.



Średnia ocena: 8,5

Obejrzany ▼

8 ▼

Dodaj [Usuń z kolekcji](#)

Wymagania pozafunkcjonalne

1. Wykorzystanie technologii Microsoft ASP.NET Core.
2. Wykorzystanie Entity Framework Core.
3. Wykorzystanie Microsoft Identity.

Wybrane wymagania funkcjonalne (przypadki użycia)

UC1: Rejestracja w serwisie

Główny aktor: Niezalogowany użytkownik

Główny scenariusz:

1. Niezalogowany użytkownik wchodzi na stronę główną z zamiarem założenia konta w serwisie.
2. Przechodzi na strony rejestracji, klikając przycisk „rejestracja” w menu górnym strony.
3. Wypełnia pola formularza: adres e-mail, nickname oraz hasło.
4. Aplikacja tworzy nowe konto i przekierowuje zalogowanego użytkownika na stronę główną.

Rozszerzenia:

3.A. Użytkownik nie wypełnił wszystkich pól lub podał dane niezgodne z wymaganiami.

3.A.1. System wyświetla informacje o nieprawidłowych/wymaganych danych.

UC2: Dodanie oceny filmu

Główny aktor: Zalogowany użytkownik

Główny scenariusz:

1. Użytkownik przechodzi na listę filmów, wybierając odpowiednią pozycję z menu górnego.
2. Z listy wybiera interesujący go film i przechodzi na jego podstronę.
3. Wybiera odpowiednią ocenę w skali 1-10 i potwierdza, klikając „ocenić”. Ocena została dodana.
4. Użytkownik przechodzi na listę swoich filmów, klikając przycisk „moje filmy” w menu głównym.
5. Oceniony film widoczny jest na liście.

Rozszerzenia:

2.A. Użytkownik wybiera odpowiednią ocenę przy filmie bezpośrednio na liście i potwierdza, klikając „ocenić”. Ocena została dodana. Przejście do 4.

UC3: Wejście na stronę bez odpowiednich uprawnień

Główny aktor: Zalogowany użytkownik o roli „użytkownik”.

Główny scenariusz:

1. Użytkownik w menu głównym serwisu nie widzi stron, do których nie ma dostępu.
2. Użytkownik wpisuje adres do niedostępnej dla niego strony w pasku adresu przeglądarki.
3. System przekierowuje użytkownika na stronę główną z informacją o braku dostępu.

UC4: Dodanie komentarza na karcie filmu

Główny aktor: Zalogowany użytkownik

Główny scenariusz:

1. Użytkownik przechodzi na kartę wybranego filmu.
2. Na dole podstrony widzi pole tekstowe z możliwością dodania komentarza, a pod nim listę dodanych wcześniej wpisów dot. Filmu.
3. Użytkownik uzupełnia pole tekstowe i zatwierdza dodanie komentarza, klikając przycisk „skomentuj”.
4. Komentarz zostaje dodany do listy.

Rozszerzenia:

3.A. Użytkownik zostawia puste pole tekstowe i zatwierdza dodanie komentarza.

3.A.1. Wyświetlona zostaje informacja o konieczności uzupełnienia treści komentarza.

UC5: Wyświetlenie profilu innego użytkownika

Główny aktor: Zalogowany użytkownik

Główny scenariusz:

1. Użytkownik, przeglądając komentarze, klika w nickname innej osoby.
2. Zostaje przeniesiony na stronę profilu, na której widzi listę ostatnich filmów danego użytkownika.

UC5: Wyświetlenie rankingu filmów

Główny aktor: Użytkownik (zalogowany bądź nie)

Główny scenariusz:

1. Użytkownik wybiera z menu głównego serwisu pozycję „ranking filmów”.
2. Zostaje przeniesiony do podstrony, na której może zobaczyć 10 filmów z najlepszą średnią ocen, a także samą średnią.

Wybrane technologie

Backend aplikacji

Technologia: ASP.NET Core

Uzasadnienie decyzji: Wybór narzucony. Poza tym framework ten pozwala na szybkie tworzenie monolitycznych aplikacji internetowych, a także API.

Potencjalne problemy: Brak.

Alternatywy: Brak.

Frontend aplikacji

Technologia: Szablony Razor

Uzasadnienie decyzji: Wybór narzucony. Ponadto szablony Razor są częścią frameworka ASP.NET Core, co zapewnia bezproblemową integrację.

Potencjalne problemy: Szablony Razor nie udostępniają tylu możliwości co osobny frontend zbudowany w jednej z technologii Javascriptowych.

Alternatywy: Brak.

System bazy danych

Technologia: Microsoft SQL Server (wybór wstępny)

Uzasadnienie decyzji: Dobra integracja z aplikacjami napisanymi z wykorzystaniem platformy .NET za pomocą Entity Framework Core.

Potencjalne problemy: Brak znajomości technologii wśród członków zespołu. Co prawda EF Core praktycznie całkowicie wyklucza konieczność pisania własnych zapytań SQL, jednak w przypadku konieczności debugowania błędów może stanowić to dodatkową trudność.

Alternatywy: PostgreSQL.

Struktura projektu

Projekt zbudowany jest w oparciu o framework ASP .NET MVC i z tego też względu posiada strukturę typową dla tego wzorca.

Cała aplikacja dzieli się na trzy części:

- Modele – klasy reprezentujące encje bazodanowe. Należą do nich:
 - o Movie
 - o Collection
 - o Genre
 - o Role
 - o MovieComment
- Widoki – pliki reprezentujące warstwę frontendową, widoczną przez użytkownika. Dzielą się głównie na:
 - o Widoki dla stron związanych z filmami
 - o Widoki filmów dla administratora
 - o Pozostałe, czyli strona główna i kokpit.
- Kontrolery – warstwa odpowiadająca za odpowiednie kierowanie akcji wykonanych przez użytkownika (np. do odpowiednich modeli). W projekcie znajdują się kontrolery:
 - o Filmów
 - o Stron domowych
 - o Zarządzania filmami przez administratora

Ponadto utworzone zostały dodatkowe przestrzenie na nowe wyliczeniowe typy danych (np. CollectionStatus, który nie jest mapowany na bazę danych), a także inne narzędzia (dodatkowy walidator weryfikujący status filmu dodanego do kolekcji).

Diagram klas

Diagram klas obrazujący relacje między encjami.

