



Politechnika Wrocławska

**Wydział Informatyki i Zarządzania**

kierunek studiów: Informatyka

specjalność: wpisz właściwą

Praca dyplomowa - inżynierska

## PROJEKT I IMPLEMENTACJA SYSTEMU DO REKOMENDACJI FILMÓW

Mateusz Brożyna

słowa kluczowe:  
serwis internetowy  
Web scraping  
system rekomendacji

krótkie streszczenie:

Niniejsza praca zawiera wszelkie informacje dotyczące procesu projektowania oraz implementacji systemu webowego, mającego na celu rekomendację filmów ze względu na kryterium wprowadzone przez użytkownika. Dodatkowo w dokumencie zawarto opis i sposób działania robota internetowego służącego do zbierania danych na potrzeby serwisu.

opiekun pracy dyplomowej	dr inż. Adrianna Kozierkiewicz-Hetmańska	.....	.....
	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do: \*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

\* niepotrzebne skreślić

pieczęćka wydziałowa

Wrocław 2016

## **Streszczenie**

Celem niniejszej pracy jest projekt i implementacja systemu do rekomendacji filmów na podstawie kryterium podanego przez użytkownika. W pierwszej części zostanie przedstawiona geneza pracy, a następnie wymagania projektowe i implementacyjne. Kolejnym elementem jest projekt prototypu aplikacji, oraz opis prac implementacyjnych przeprowadzonych na podstawie tegoż projektu. Ostatni etap pracy zawiera opisy przeprowadzonych testów i podsumowanie przeprowadzonych prac wraz z analizą implementacji i sugestiami dotyczącymi możliwości dalszego rozwoju projektu. Realizacja projektu została oparta o środowisko MEAN.JS i język JavaScript oraz nierelacyjny system baz danych NoSQL MongoDB.

## Spis Treści

Streszczenie .....	2
Wstęp.....	6
1.1 Wprowadzenie do problematyki .....	6
1.2 Geneza pracy .....	6
1.3 Cel i zakres pracy .....	6
Wymagania projektowe.....	7
Stan wiedzy i techniki w zakresie tematyki pracy .....	8
3.1 Wprowadzenie.....	8
3.2 Przegląd podobnych istniejących rozwiązań.....	8
3.3 Podsumowanie i wnioski.....	9
Założenia projektowe .....	10
4.1 Przedmiot prac.....	10
4.2 Wymagania funkcjonalne.....	10
4.3 Wymagania нефункционалне.....	11
4.4 Opis podstawowej architektury systemu .....	11
4.5 Sposób realizacji .....	11
Projekt .....	13
5.1 Wprowadzenie.....	13
5.2 Przypadki użycia .....	13
5.3 Interfejs.....	20
5.4 Baza danych .....	21
5.4.1 Model logiczny.....	21
5.4.2 Model fizyczny.....	24
Implementacja systemu do rekomendacji filmów .....	25
6.1 Wprowadzenie.....	25
6.2 Wykorzystane środowiska i narzędzia programistyczne.....	25
6.3 Architektura wysokiego poziomu .....	26
6.4 Wyszukiwarka filmów .....	35
6.5 Web Scraper .....	37
Testy .....	39
7.1 Testy poprawności działania .....	39
7.2 Testy jednostkowe i testy bezpieczeństwa .....	39
7.3 Podsumowanie .....	40
Podsumowanie pracy.....	41
Bibliografia.....	42

## Spis rysunków

Rysunek 5.1 Diagram przypadków użycia.....	14
Rysunek 5.2 REQ001: PU001 - Logowanie do serwisu – diagram aktywności.....	15
Rysunek 5.3 REQ002: PU002 - Rejestracja użytkownika – diagram aktywności.....	16
Rysunek 5.4 REQ002.2, REQ002.3: PU003 - Edycja danych użytkownika, PU004 - Usuwanie użytkownika - Diagram aktywności.....	17
Rysunek 5.5 REQ006: PU006 – Wyszukiwanie filmów– Diagram aktywności .....	18
Rysunek 5.6 REQ009: PU007 – Uruchomienie Scrappera– Diagram aktywności.....	19
Rysunek 5.7 REQ007 – Historia wyszukiwania w serwisie – Diagram aktywności .....	20
Rysunek 5.8 Diagram przepływu sterowania.....	21
Rysunek 5.9 Model logiczny bazy danych.....	22
Rysunek 6.1 Logowanie użytkownika – widok użytkownika.....	26
Rysunek 6.1.1 Logowanie użytkownika (błąd logowania) – widok użytkownika .....	26
Rysunek 6.2 Rejestracja użytkownika – widok użytkownika.....	27
Rysunek 6.2.2 Rejestracja użytkownika (walidacja danych) – widok użytkownika .....	27
Rysunek 6.3 Lista filmów – widok użytkownika.....	28
Rysunek 6.4 Szczegóły użytkownika – widok użytkownika .....	29
Rysunek 6.5 Wyszukiwarka – widok użytkownika .....	29
Rysunek 6.6 Scraper – widok administratora.....	30
Rysunek 6.7 Szczegóły filmu – widok użytkownika .....	30
Rysunek 6.8 Lista użytkowników– widok administratora .....	31
Rysunek 6.9 Ustawienia profilu (edycja danych) – widok użytkownika.....	31
Rysunek 6.10 Ustawienia profilu (zmiana zdjęcia profilowego) – widok użytkownika .....	32
Rysunek 6.11 Ustawienia profilu (zmiana hasła) – widok użytkownika.....	32
Rysunek 6.12 Zakaz dostępu.....	33
Rysunek 6.13 Nie znaleziono strony .....	33
Rysunek 6.14 Historia wyszukiwania – widok użytkownika.....	34
Rysunek 6.15 Historia wyszukiwania (brak historii) – widok użytkownika .....	34
Rysunek 6.16 Szczegóły wyszukiwania – widok użytkownik.....	35
Rysunek 7.1 Wynik badania szybkości działania aplikacji.....	39

## Spis tabel

Tabela 5.1 Szczegóły encji histories .....	23
Tabela 5.2 Szczegóły encji films.....	23
Tabela 5.3 Szczegóły encji users.....	24

# Rozdział 1

## Wstęp

### 1.1 Wprowadzenie do problematyki

W dzisiejszych czasach komputer jest głównym źródłem informacji. Codziennie używany jest jako narzędzie służące do znajdowania odpowiednich, polecanych dla gustu użytkownika filmów. Przeglądanie sieci daje podstawy by sądzić, że każdy internauta chce używać Internetu w spersonalizowany, przystosowany dla siebie sposób. Dlatego też ruch w sieci jest stale monitorowany, a dane zbierane są po to, żeby by móc lepiej wybierać źródła dla użytkownika. Problem w tym, że taka personalizacja w wielu przypadkach wymaga posiadania kilku kont, w wielu różnych serwisach, które takie dane zbierają. Wychodząc temu naprzeciw powstała idea narzędzia pozwalającego na skrócenie czasu rejestracji oraz personalizacji klienta w serwisach filmowych co w następstwie ograniczyłoby czas poświęcony na znalezienie odpowiedniego filmu. Stąd też inicjatywa powstania takiego serwisu, a jego mottem przewodnim były słowa: „By godziny spędzone na znalezienie filmu zamienić w minuty”.

### 1.2 Geneza pracy

Podstawą do utworzenia przedstawionego systemu, był czas poświęcony na szukanie odpowiedniego filmu godnego rekomendacji. Rozwiązania dostępne na rynku nie zaspokajały oczekiwań i potrzeb użytkowników, a serwisy na podstawie których system został utworzony udostępniały zbyt ograniczone możliwości wyszukiwania – szerzej opisane w podrozdziale „Przegląd podobnych istniejących rozwiązań”. Użytkownicy potrzebowali możliwości skonfrontowania poszczególnych filmów z opiniami innych użytkowników (także zagranicznych). Różnicę w opiniowaniu, dzięki tej implementacji można zobaczyć w rozbieżnościach rankingów poszczególnych źródeł, co tylko podkreśla słuszność podstaw do stworzenia takiej wyszukiwarki.

### 1.3 Cel i zakres pracy

Celem pracy jest wykonanie projektu i implementacji wyszukiwarki filmowej na podstawie podanych przez użytkownika kryteriów. Zakres prac tego dokumentu obejmuje wprowadzenie do problematyki, genezę powstania pracy(1.2), jej cel i zakres (1.3). W 2 rozdziale omówione zostały wymagania projektowe. Rozdział 3 zawiera informacje dotyczące stanu istniejących rozwiązań (3.2) oraz technik przydatnych w realizacji pracy(3.3). Kolejny rozdział to przedstawienie założeń projektowych: przedmiotu prac(4.1), wymagań funkcjonalnych(4.2), niefunkcjonalnych (4.3) oraz opisu architektury systemu (4.4) i sposobu realizacji (4.5). Rozdział 5 zawiera informacje dotyczące projektu aplikacji: przypadków użycia(5.2), interfejsu(5.3) i bazy danych(5.4). Kolejny rozdział to omówienie implementowanego prototypu aplikacji: jej środowiska i narzędzi (6.2), interfejsów (6.2) oraz funkcji (6.4) i (6.5). Rozdział 7 zawiera opis przeprowadzonych testów podzielonych ze względu na zakres działań na testy poprawności działania (7.1), jednostkowe i bezpieczeństwa (7.2). Ostatni rozdział zawiera informacje dotyczące wykonanych prac, realizacji celu pracy, analizy przeprowadzonych działań i sugestie odnośnie dalszych prac .

## Rozdział 2

### Wymagania projektowe

Główną funkcjonalnością aplikacji ma być możliwość wyszukiwania filmów na podstawie podanych przez użytkownika kryteriów takich jak:

- długość filmu
- reżyser
- aktorzy
- serwis źródłowy
- gatunek filmu
- kraj produkcji
- rok produkcji

Dodatkowo system ma przechowywać informacje dotyczące historii wyszukiwania poszczególnych użytkowników, umożliwiać przegląd bazy filmów, oraz ich filtrowanie na podstawie przedstawionych wyżej kryteriów. Po stronie administratora serwis ma dawać możliwość ręcznego uruchamiania robota internetowego do zbierania źródeł danych. Implementacja ma zapewniać możliwość zarządzania użytkownikami w panelu administratora tzn. edycji danych i usuwania użytkowników.

Docelowym użytkownikiem aplikacji mają być wszyscy, którzy potrzebują wyszukiwarki filmowej umożliwiającej skonfrontowanie opinii z poszczególnych stron źródłowych. Projekt jest serwisem webowym, zatem możliwość korzystania z tej wyszukiwarki zapewnia każda dostępna przeglądarka internetowa. Jeśli chodzi o środowisko w jakim aplikacja ma działać elementem wystarczającym będzie dowolny serwer WWW np. Nginx. Dodatkowym atutem byłaby możliwość instalacji własnej bazy danych MongoDB, nie jest to jednak warunek konieczny, gdyż istnieje wiele serwisów udostępniających wyżej wymienione bazy danych bezpłatnie.

## Rozdział 3

### Stan wiedzy i techniki w zakresie tematyki pracy

#### 3.1 Wprowadzenie

W niniejszym rozdziale zostanie przedstawione podsumowanie dotyczące istniejących rozwiązań i ich wpływ na realizowaną aplikację. Pierwszym punktem będzie przegląd oraz analiza istniejących rozwiązań skupiająca się na wadach i zaletach serwisów, zwracająca także uwagę na różnice pomiędzy prezentowanymi rozwiązaniami a projektem realizowanym w tej pracy. Następnym punktem będzie krótkie podsumowanie wraz z wnioskami - informacjami jak przegląd rozwiązań wpłynie lub powinien wpłynąć na realizowany projekt.

#### 3.2 Przegląd podobnych istniejących rozwiązań

Przegląd istniejących rozwiązań zacznie się od krótkiego przedstawienia serwisów, na podstawie których implementacja będzie tworzona. Ponieważ realizowana aplikacja będzie bazą filmową, przegląd będzie skupiał się na tym szczególnym elemencie prezentowanych serwisów.

Pierwszym z nich jest filmweb.pl [7]. Udostępnia on własną bazę filmową, na podstawie której użytkownicy mogą komentować oraz oceniać poszczególne filmy, oceniać aktorów oraz inne osoby biorące udział przy produkcji filmu (jeśli zostały uwzględnione w bazie). Możliwości wyszukiwarki serwisu ograniczają się do wyszukiwania jedynie po nazwie (filmu, aktora). Warto nadmienić również ogromny rozmiar bazy filmowej serwisu, która obecnie wynosi ponad 500.000 filmów co czyni ją największą internetową bazą w Polsce i drugą co do wielkości na świecie po imdb.com.

Imdb.com „Internet Movie Database” [9] - największa internetowa baza filmów, której właścicielem jest firma Amazon. Zawiera ponad 2.700.000 rekordów, z czego ponad 500.000 to filmy pełnometrażowe. Podobnie jak filmweb.com zawiera dane dotyczące filmów, aktorów oraz ludzi związanych z kinem. Zarejestrowani użytkownicy oceniają filmy, a na ich podstawie tworzone są rankingi popularności. Serwis używa algorytmu spersonalizowanej rekomendacji na podstawie ocenionych i obserwowanych filmów.

Serwis filmaster.pl [6] jest najmniejszym z przedstawionych serwisów, pomimo, że sam chwali się posiadaniem ponad 100 000 użytkowników. Zawiera newsy dotyczące filmów, kinowe nowości, ale jednocześnie ranking w którym mamy możliwość wyszukiwania filmów, aktorów i ludzi filmu na podstawie wybranych tagów. Ciekawym rozwiązaniem w tej aplikacji są rankingi filmowe podzielone na kilka kategorii np. reżyseria, scenariusz, aktorstwo, montaż, muzyka. Mamy dzięki temu możliwość sortowania bazy ze względu na oceny w poszczególnym rankingu. Największym problemem prezentowanego serwisu jest brak spójności danych w porównaniu do wcześniej prezentowanych aplikacji. W wielu przypadkach brakuje szczegółowych informacji takich jak oryginalne tytuły czy opisy filmów. Kolejnym poważnym mankamentem zdaje się być rozmiar filmowej bazy danych, który wg szacunków przedstawionych na stronie zawiera niecałe 800 rekordów.

Suggestmemovie.com [8] jest to aplikacja znacząco podobna do mającego powstać projektu. Jest to serwis używający bazy imdb.com do filtrowania wyników i prezentowania oczekiwanych przez użytkownika filmów np. ze względu na minimalną ocenę, gatunek filmowy, rok produkcji, liczbę głosów lub słowa kluczowe. Sądząc po informacjach zawartych



na stronie do zbierania danych używa ona techniki Web scrapingu po źródłach imdb.com. Największą wadą serwisu jest jednak praktycznie stuprocentowe odwzorowanie wyszukiwarki IMDb, co w ocenie użytkowników czyni go bezużytecznym.

Ostatnim wartym uwagi elementem będzie przedstawienie używanych obecnie rozwiązań służących do polecania filmów. Jednym z najpopularniejszych obecnie stosowanych algorytmów rekomendacji jest tzw. „collaborative filtering” [5]. Jest to metoda dostarczająca opinii użytkownikowi badając jego podobieństwa z innymi klientami serwisu, lub naszymi ocenami. Serwisy takie jak filmweb.pl czy Netflix stosują ją na zasadzie ocen i poleceń użytkownika, a na ich podstawie generowane są filmy polecane dla klienta. Algorytmy rekomendacji są zagadnieniem szeroko rozwijanym. Wiele obecnie prowadzonych badań dąży do stworzenia idealnego i uniwersalnego schematu. Działanie algorytmu dzieli się na dwie główne gałęzie:

- Item-based system – prognozy dla użytkownika tworzone są na podstawie historii działań użytkownika. Więcej o tej metodzie i przeprowadzonych przy jej użyciu symulacjach można przeczytać w artykule „Item-based Collaborative Filtering Algorithm”[4].
- User-based system – prognozy dla użytkownika tworzone są na podstawie macierzy najbliższego sąsiedztwa. Działania użytkowników sytuują ich na tej macierzy, a znajdujące się „najbliżej” elementy można uznać za te o podobnych gustach. Opis i zastosowanie algorytmu przy użyciu chmury obliczeniowej zostały przedstawione w publikacji „User-Based Collaborative-Filtering Recommendation System on Hadoop” [11].

Szerszy opis samego algorytmu, oraz ciekawe rozwiązanie łączące przedstawioną metodę z macierzą faktoryzacji i algorytmem najbliższego sąsiedztwa zostały przedstawione w publikacji [2].

### **3.3 Podsumowanie i wnioski**

Żadne z przedstawionych w tym rozdziale istniejących rozwiązań nie spełnia wymagań dotyczących tworzonej aplikacji. By móc konkurować pod względem technicznym z wymienionymi serwisami wymagana będzie realizacja niektórych funkcjonalności udostępnianych przez te aplikacje. Pierwszą z nich z pewnością będzie prezentacja bazy filmowej oraz wyszukiwarka filmowa umożliwiająca filtrowanie bazy na podstawie różnych kryteriów np. reżyser, tytuł filmu, rok produkcji, aktor, tytuł oryginalny, typ, kraj. Głównym punktem tworzonego projektu nadal będzie jednak wyszukiwarka filmowa oraz system zbierający dane. Będzie to najważniejszy element podczas projektowania oraz implementacji serwisu.

## Rozdział 4

### Założenia projektowe

#### 4.1 Przedmiot prac

Przedmiotem pracy jest wykonanie projektu wyszukiwarki filmowej na podstawie podanych przez użytkownika kryteriów.

#### 4.2 Wymagania funkcjonalne

- REQ001** System daje możliwość zalogowanie się za pomocą loginu użytkownika i hasła
- REQ002** System umożliwia rejestrację użytkownika po podaniu imienia, nazwiska, unikatowego adresu email, unikatowej nazwy użytkownika oraz hasła spełniającego zadane kryteria.
  - REQ002.1** System nadaje role zarejestrowanym użytkownikom. Po rejestracji użytkownikowi nadawana jest rola user, która może być edytowana przez użytkownika z rolą admin.
  - REQ002.2** System daje możliwość edycji danych użytkownika (poza adresem email) przez użytkowników ze zdefiniowaną rolą admin.
  - REQ002.3** System umożliwia usuwanie zarejestrowanych kont dla użytkownika z rolą admin.
- REQ003** System umożliwia przeglądanie bazy filmów
- REQ004** System umożliwia filtrowanie bazy filmów po dowolnych danych umieszczonych w bazie danych
- REQ005** System daje możliwość wyszukiwania filmu po podanych przez użytkownika kryteriach
- REQ006** System daje możliwość przeglądania historii wyszukiwania użytkownika dla zalogowanych użytkowników bez względu na rolę
- REQ007** System daje możliwość przeglądania historii wyszukiwania w serwisie dla użytkowników zarejestrowanych i niezarejestrowanych
- REQ008** System daje możliwość uruchomienia automatycznego robota internetowego do zbierania danych dla użytkowników z rolą admin
- REQ009** System daje możliwość przeglądania szczegółowych danych dotyczących bazy filmów dostępnych w panelu administratora dla użytkowników z rolą admin

### 4.3 Wymagania niefunkcjonalne

- REQ001** Hasła użytkowników w serwisie są przechowywane w sposób zaszyfrowany przy użyciu kodowania Base64 i dodanie ciągu zaburzeń.
- REQ002** Uruchomienie aplikacji na środowisku developerskim wymaga instalacji Git, Grunt, Karma, Gulp, Bower oraz MongoDB.
- REQ003** Do instalacji aplikacji na serwerze wymagane jest stworzenie posiadanie serwera WWW np. Nginx. W przypadku korzystania z zewnętrznej bazy danych nie ma konieczności instalacji bazy danych MongoDB na serwerze.
- REQ004** Kolekcje w bazie tworzone są automatycznie po pierwszym dodaniu danych. Do pełnej funkcjonalności wymagane są kolekcje przechowujące dane użytkowników, sesji, filmów oraz historii wyszukiwania.
- REQ005** Pierwsza instalacja aplikacji wymaga utworzenia użytkownika z rolą admin.
- REQ006** Uruchomienie aplikacji po stronie klienta wymaga jedynie przeglądarki z dostępem do Internetu oraz obsługą JavaScript.

### 4.4 Opis podstawowej architektury systemu

Aplikacja działać będzie w architekturze klient – serwer. Do poprawnego działania wymaga instalacji na zewnętrznym serwerze WWW oraz przeglądarki z dostępem do Internetu po stronie klienta.

### 4.5 Sposób realizacji

Do przygotowania aplikacji webowej zostanie użyte narzędzie MEAN.JS. Jest to rozwiązanie typu „open-source” bazujące na JavaScript służące do budowania aplikacji webowych. Nazwa MEAN jest akronimem od wyrazów MongoDB, Express, AngularJS oraz Node.js:

- MongoDB – nierelacyjna baza danych, zapewniająca dużą wydajność oraz skalowalność. Aplikacja przechowywać będzie zebrane informacje o poszczególnych filmach, dane użytkowników, oraz historie ich wyszukiwania.
- Express – Framework Node.js, część serwerowa stosu.
- AngularJS – Framework Front-end [1] JavaScript stworzony przez Google umożliwiający tworzenie dynamicznych stron internetowych używając własnego kompilatora HTML
- Node.js – środowisko uruchomieniowe dla kodu JavaScript po stronie serwera, stosowanie podczas tworzenia kodu.

Szczegółowe opisy rozwiązania [3], oraz dokładne omówienie poszczególnych elementów dostępne są na oficjalnych stronach producentów oraz github.com. Dodatkowymi elementami aplikacji będzie użycie Bootstrap, który często nazywany jest frameworkiem CSS zapewniającym responsywność i łatwość w tworzeniu interfejsu użytkownika dla stron internetowych.

Szkielet projektu zostanie utworzony przy użyciu generatora MEAN.JS narzędzia Yeoman [10]. Pozwoli to na zachowanie najlepszych praktyk programistycznych przy rozwijaniu projektu, oraz ułatwi początkowe prace związane z instalacją komponentów i ich połączeń. Dodatkowo pozwoli na zachowanie wysokiej wydajności. Do utworzenia tego projektu wymagana będzie instalacja następujących pakietów:

- Node
- Node Package Manager
- Git
- Grunt
- Karma
- Gulp
- Bower
- MongoDB

Następnym elementem będzie import projektu do środowiska programistycznego WebStorm, które służyć będzie jako narzędzie do rozwijania aplikacji o nowe funkcjonalności, a także testowania aplikacji po stronie serwera.

Testowanie aplikacji po stronie klienta będzie się odbywać przy użyciu standardowych przeglądarek internetowych:

- Google Chrome wersja: 50.0.2661.102 m
- Mozilla Firefox wersja: 46.0.1
- Internet Explorer wersja: 11.0.9600.18321

Zarządzanie bazą danych odbywać się będzie przy użyciu narzędzia Robomongo 0.9.0. Dodatkowo w celach bezpieczeństwa i ułatwienia pracy użyty będzie system kontroli wersji git i program GitHub / GitShell. Projekt aplikacji zostanie zrealizowany przy użyciu programu Visual Paradigm wersja 13.1.

## **Rozdział 5**

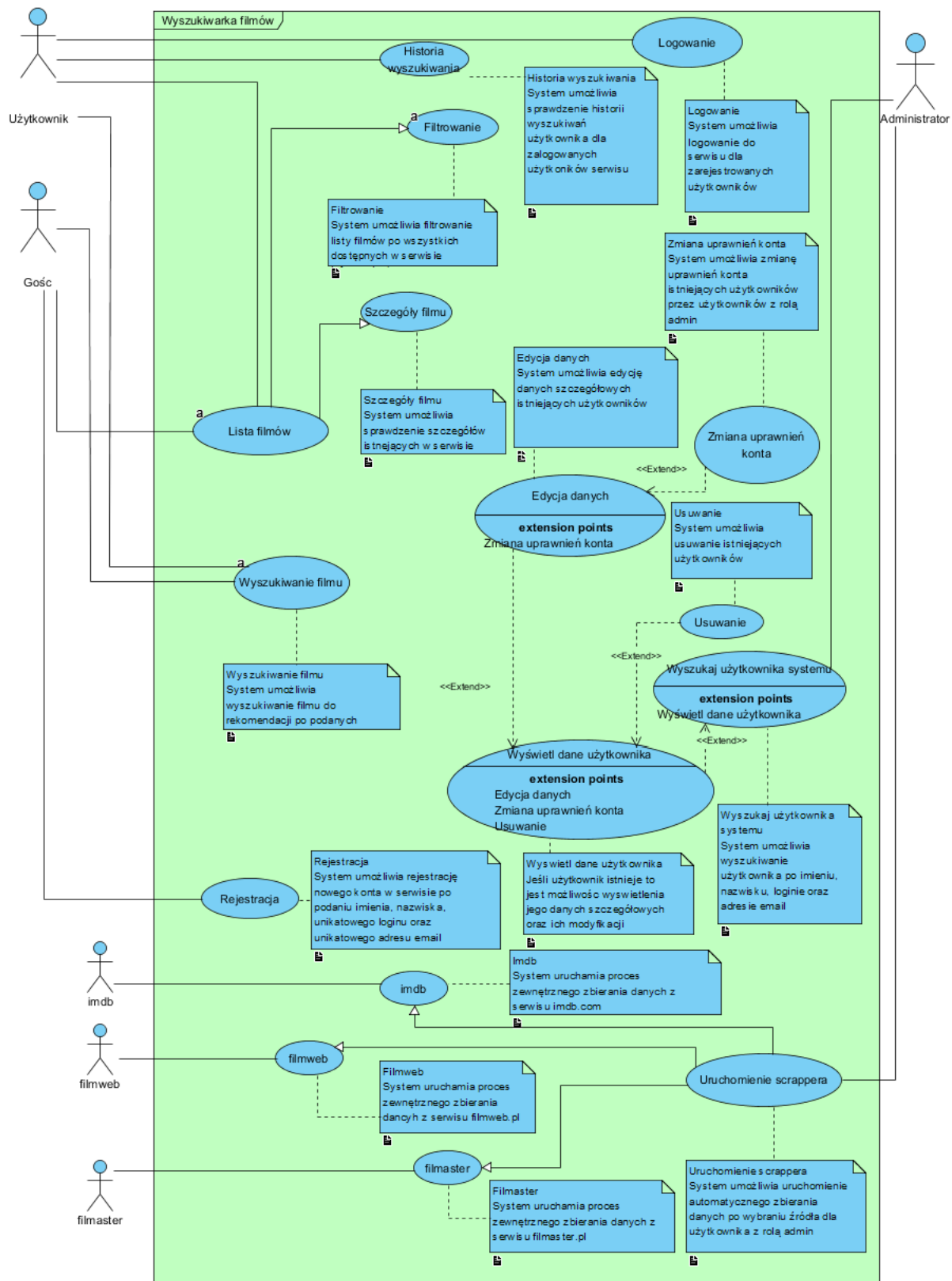
### **Projekt**

#### **5.1 Wprowadzenie**

W tym rozdziale zostanie przedstawiony projekt aplikacji. Pierwszym etapem będzie przedstawienie diagramu przypadków użycia, zwracające uwagę na zdefiniowane wcześniej wymagania funkcjonalne. Następnym elementem będą szczegółowe opisy poszczególnych przypadków użycia, prezentujące ich scenariusze główne i alternatywne oraz diagramy aktywności użytkownika. Kolejnym etapem będzie przedstawienie diagramu interfejsów oraz schematu bazy danych, na której opierał będzie się realizowany system. Jest to o tyle ważne, że w przypadku tego projektu dobry model bazy danych pozwoli uniknąć w przyszłości jej modyfikacji. Dodatkowo w związku z możliwością rozszerzenia bazy danych przy użyciu Web scrapingu poprawnie zaprojektowany i zrealizowany model bazy zoptymalizuje szybkość zapytań.

#### **5.2 Przypadki użycia**

By zachować przejrzystość prezentowanych przypadków użycia, pierwszym elementem będzie przedstawienie ogólnego ich diagramu w realizowanej aplikacji. Następnie prezentacja poszczególnych przypadków użycia zgodnie z wymaganiami funkcjonalnymi, przedstawionymi w podrozdziale „Wymagania funkcjonalne”. Przypadki użycia, które nie wymagają opisywania w postaci diagramu aktywności użytkownika (prezentowane w konkretnym miejscu aplikacji bez potrzeby interakcji z użytkownikiem) zostały umieszczone tylko w diagramie przypadków użycia i scenariuszach przypadków użycia.



Rysunek 5.1 Diagram przypadków użycia

## PU001 Logowanie do serwisu

*Zdarzenie inicjujące:* Wybranie opcji zaloguj się w panelu menu

*Warunki początkowe:* Brak

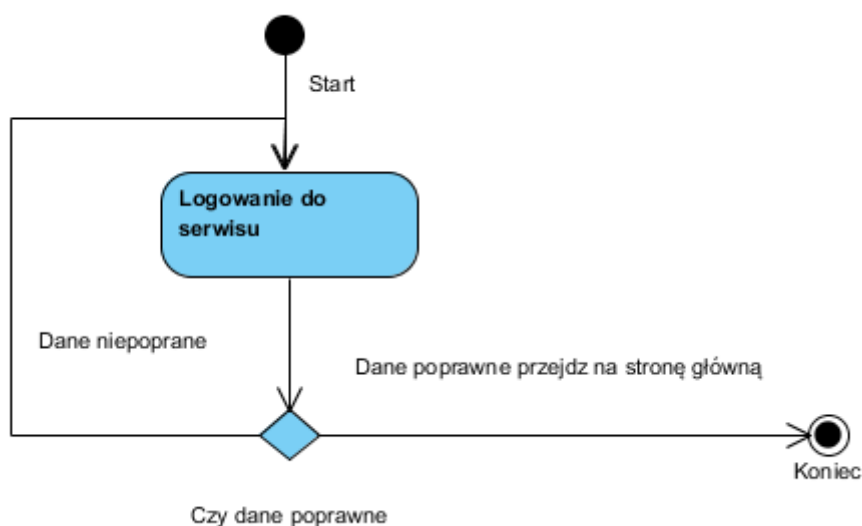
*Opis przebiegu interakcji:*

1. Użytkownik podaje login oraz hasło do serwisu oraz wciska przycisk logowania
2. System autoryzuje użytkownika
3. System przekierowuje użytkownika na stronę główną aplikacji.

*Przebieg alternatywny:*

- 1a. Dowolna dana podana przez użytkownika jest niepoprawna
  - 1a1. System wyświetla błąd logowania

*Warunki końcowe:* Brak



Rysunek 5.2 REQ001: PU001 - Logowanie do serwisu – diagram aktywności

## PU002 Rejestracja użytkownika w serwisie

*Zdarzenie inicjujące:* Wybranie opcji zarejestruj się w panelu menu

*Warunki początkowe:* Użytkownik niezalogowany

*Opis przebiegu interakcji:*

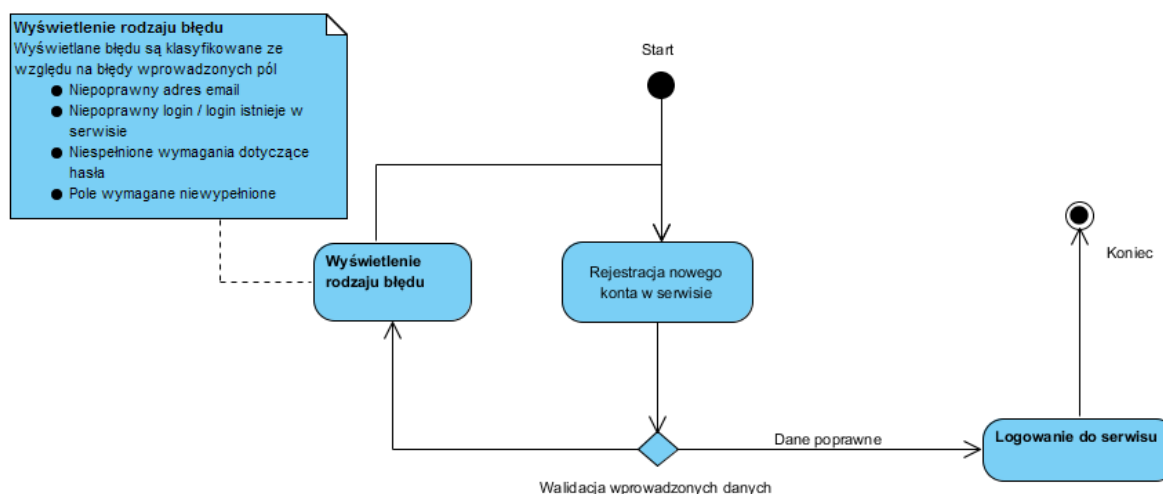
1. Użytkownik podaje imię, nazwisko, email, login oraz hasło do serwisu i wciska przycisk zarejestruj
2. System dodaje użytkownika do bazy danych
3. System przekierowuje użytkownika na stronę główną aplikacji.

*Przebieg alternatywny:*

- 1a. Dowolna dana podana przez użytkownika jest niepoprawna lub nie spełnia wymagań

- 1a1. System wyświetla błąd rejestracji  
 2a. Dane istnieją w systemie – system wyświetla błąd rejestracji

*Warunki końcowe:* Brak



Rysunek 5.3 REQ002: PU002 - Rejestracja użytkownika – diagram aktywności

### PU003 Edycja danych użytkownika

*Zdarzenie inicjujące:* Wybranie użytkownika w panelu administracyjnym

*Warunki początkowe:* Użytkownik zalogowany z prawami administratora

*Opis przebiegu interakcji:*

1. Użytkownik podaje imię, nazwisko, oraz rolę użytkownika do serwisu i wciska przycisk update
2. System edytuje dane użytkownika
3. System przekierowuje użytkownika na stronę panelu administracyjnego.

*Przebieg alternatywny:*

- 1a. Dowolna dana podana przez użytkownika jest niepoprawna lub nie spełnia wymagań
  - 1a1. System wyświetla błąd edycji

*Warunki końcowe:* Brak

### PU004 Usunięcie użytkownika

*Zdarzenie inicjujące:* Wybranie użytkownika w panelu administracyjnym

*Warunki początkowe:* Użytkownik zalogowany z prawami administratora

*Opis przebiegu interakcji:*

1. Użytkownik wybiera opcję usunąć użytkownika.
2. System pyta czy na pewno usunąć użytkownika.
3. Użytkownik potwierdza wybór.

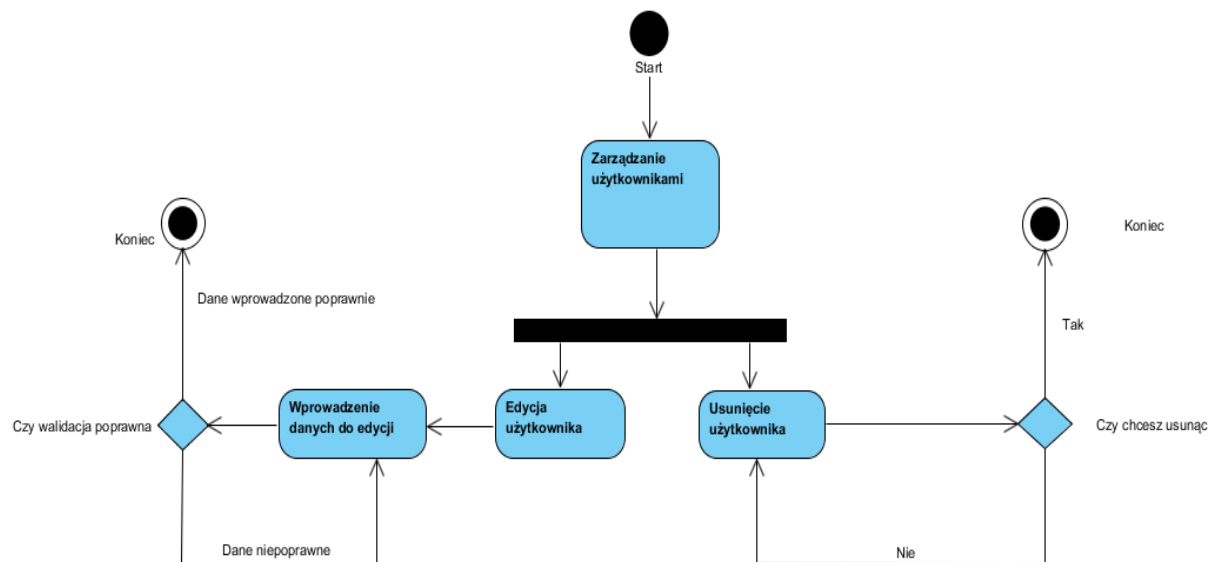


*Przebieg alternatywny:*

3a. Administrator anuluje usuwanie użytkownika.

3a1. System przekierowuje na stronę informacji o użytkowniku.

*Warunki końcowe:* Brak



Rysunek 5.4 REQ002.2, REQ002.3: PU003 - Edycja danych użytkownika, PU004 - Usuwanie użytkownika - Diagram aktywności

## PU006 Wyszukiwanie filmu

*Zdarzenie inicjujące:* Brak

*Warunki początkowe:* Brak

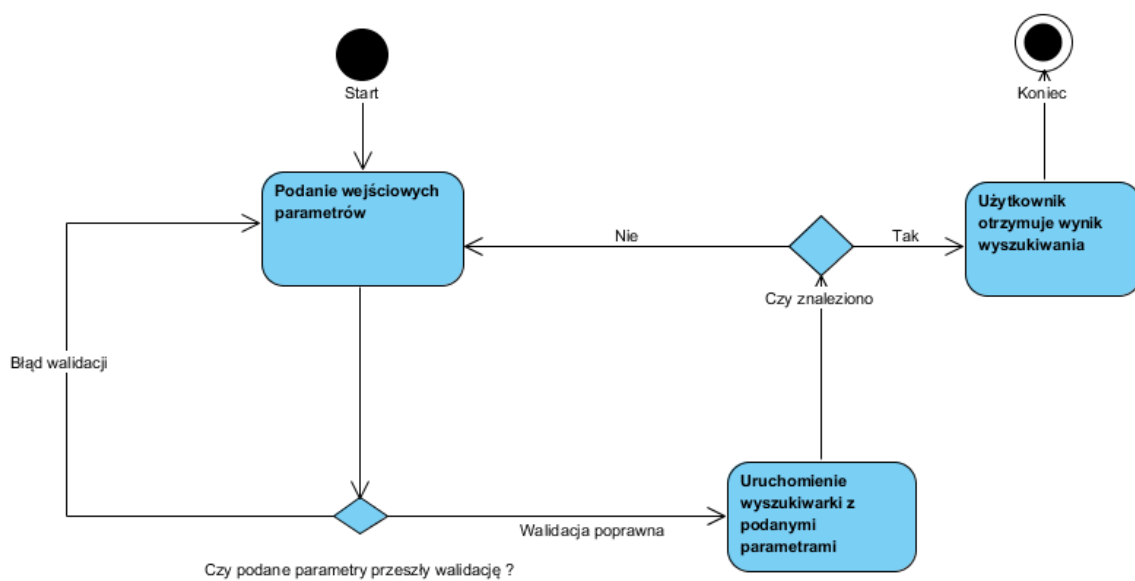
*Opis przebiegu interakcji:*

1. Użytkownik podaje dane do wyszukiwania i uruchamia wyszukiwarkę
2. System wyszukuje filmy z podanymi parametrami
3. System dodaje parametry i znaleziony film do historii wyszukiwania
4. System przekierowuje na stronę znalezionego dla użytkownika filmu

*Przebieg alternatywny:*

- 3a. System nie znalazł filmy z podanymi parametrami
- 3a1. System wyświetla błąd wyszukiwania

*Warunki końcowe:* Brak



Rysunek 5.5 REQ006: PU006 – Wyszukiwanie filmów– Diagram aktywności

## PU007 Uruchomienie Scrappera

*Zdarzenie inicjujące:* Brak

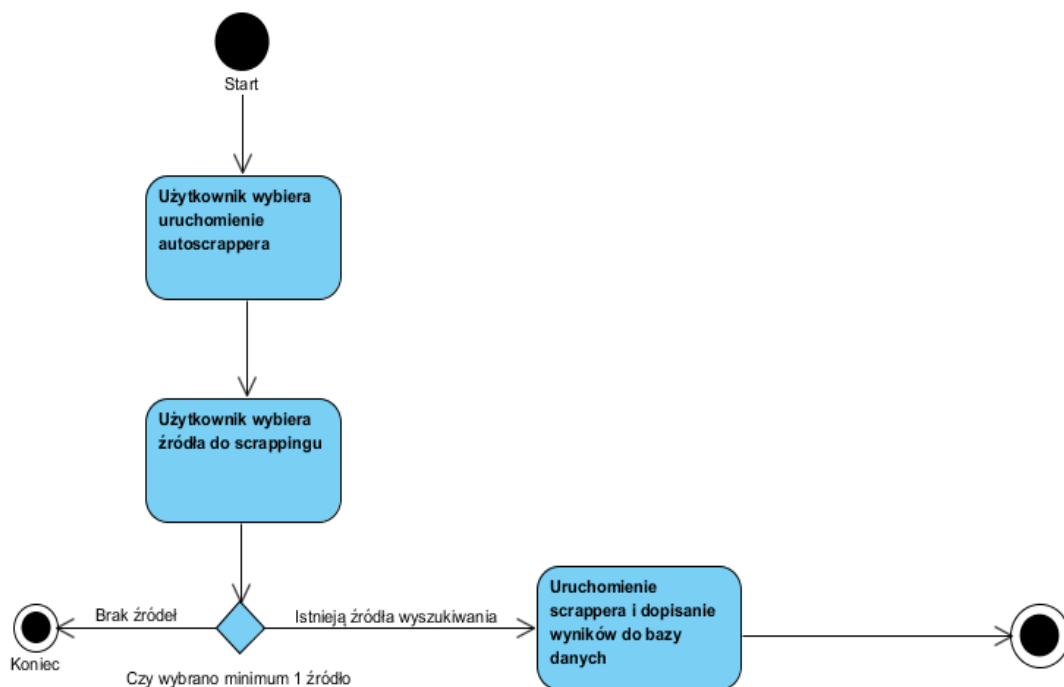
*Warunki początkowe:* Użytkownik zalogowany z prawami administratora

*Opis przebiegu interakcji:*

1. Użytkownik podaje strony źródłowe do wyszukiwania i uruchamia wyszukiwarę
2. System wyszukuje filmy z podanych stron źródłowych
3. System dodaje informacje o filmie do bazy danych

*Przebieg alternatywny:* Brak

*Warunki końcowe:* System znalazł zdefiniowaną liczbę filmów



Rysunek 6

Rysunek 5.6 REQ009: PU007 – Uruchomienie Scrappera– Diagram aktywności.

## PU008 Lista filmów

*Zdarzenie inicjujące:* Brak

*Warunki początkowe:* Użytkownik zalogowany prawami administratora

*Opis przebiegu interakcji:*

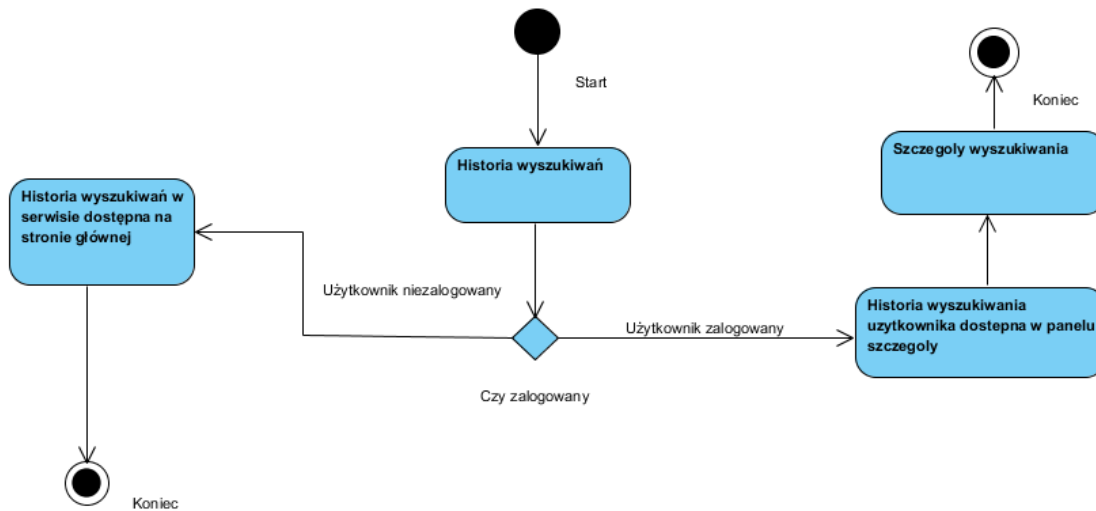
1. System prezentuje listę filmów dostępnych w systemie
2. Użytkownik wybiera film z listy wyników
3. System wyświetla szczegóły wybranego filmu

*Przebieg alternatywny:*

- 1a. Użytkownik wprowadza filtry na listę filmów
- 1a1. System filtruje wynikową listę filmów

*Warunki końcowe:* Brak

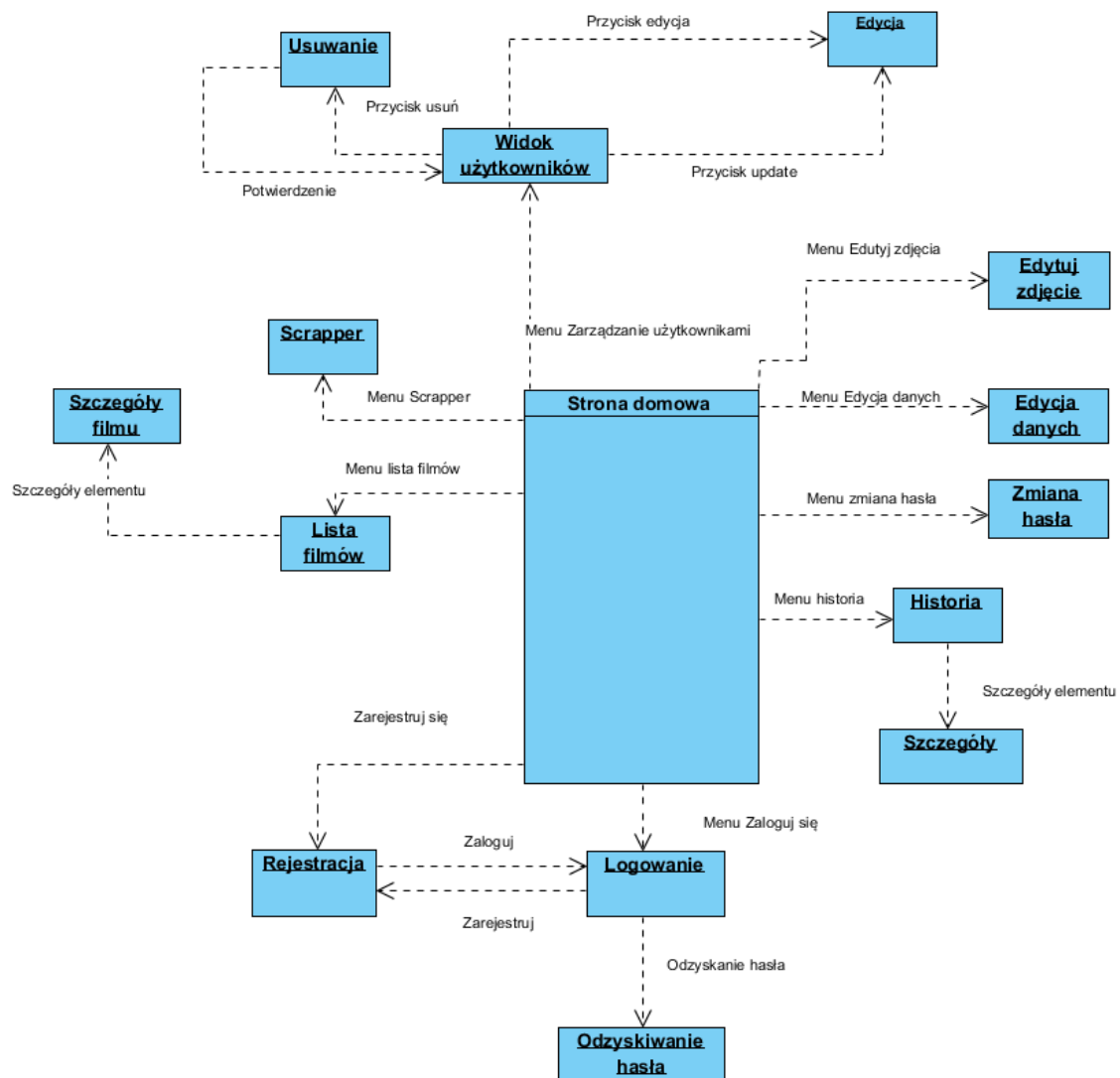
## PU009 Historia wyszukiwania



Rysunek 5.7 REQ007 – Historia wyszukiwania w serwisie – Diagram aktywności

## 5.3 Interfejs

Niniejszy podrozdział skupia się na przedstawieniu przepływu sterowania aplikacji ze względu na aktywność użytkownika. Jeden obiekt na poniższym diagramie będzie odpowiadał jednemu widokowi aplikacji. Na połączeniach pomiędzy nimi zostały przedstawione konieczne do działania aktywności. Dodatkowo z każdego miejsca aplikacji będzie można wrócić do strony domowej.



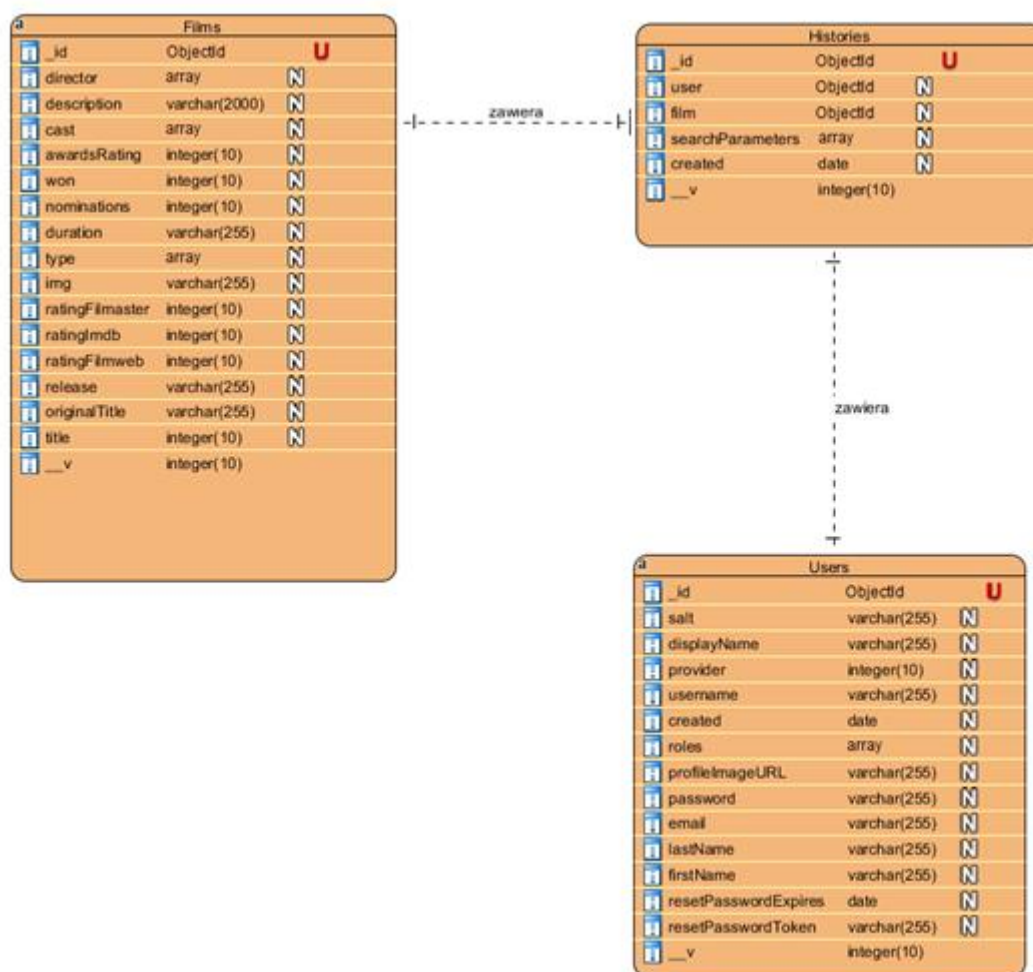
Rysunek 5.8 Diagram przepływu sterowania

## 5.4 Baza danych

Z racji użycia w projekcie nierelacyjnej bazy danych, model logiczny zostanie przedstawiony w postaci diagramu encji. Relacje pomiędzy elementami będą pilnowane po stronie aplikacji.

### 5.4.1 Model logiczny

Diagram encji przechowywanych w bazie danych kolekcji:



Rysunek 5.9 Model logiczny bazy danych

Z powodu luźnej formy przechowywania elementów w bazie MongoDB to aplikacja będzie nadzorować określone na rysunku 5.9 typy i relacje. Z powodu tak zdefiniowanych kolekcji w bazie danych wszystkie pola poza generowanymi „\_id” oraz „\_\_v” mogą być puste. Kolejnym krokiem będzie przedstawienie poszczególnych encji zaprezentowanych na diagramie:

Kolekcja histories zawiera dane dotyczące uruchomionego wyszukiwania. Umożliwia prezentowanie zalogowanemu użytkownikowi jego historii wyszukiwania, oraz ostatnich znalezionych filmów w serwisie dla niezalogowanych użytkowników.

Atrybut	Opis
__id	Automatycznie generowany numer jednoznacznie identyfikujący poszczególne wyszukiwanie
user	Parametr __id z kolekcji users identyfikujący użytkownika, który uruchomił wyszukiwarę.

film	Parametr __id z kolekcji films identyfikujący film znaleziony poprzez uruchomienie wyszukiwarki.
searchParameters	Parametry z którymi wyszukiwarka została uruchomiona. Zawiera następujące dane: <ul style="list-style-type: none"> <li>• Aktor</li> <li>• Lista krajów</li> <li>• Rok produkcji od</li> <li>• Rok produkcji do</li> <li>• Reżyser</li> <li>• Maksymalna długość</li> <li>• Serwisy na podstawie których uruchomiono wyszukiwanie</li> <li>• Minimalna ocena</li> <li>• Gatunek filmu</li> </ul>
created	Data wyszukiwania
__v	Automatyczny parametr wersji rekordu nadawany przez bazę danych

Tabela 5.1 Szczegóły encji histories

Kolekcja films zawiera zebrane informacje z serwisów Filmweb.pl, imdb.com, filmaster.pl. Umożliwia działanie wyszukiwarki, listy filmów oraz przechowywanie historii wyszukiwania.

Atrybut	Opis
__id	Automatycznie generowany numer jednoznacznie identyfikujący poszczególny film
director	Lista reżyserów filmu
description	Opis filmu
country	Lista krajów produkcji filmu
cast	Lista aktorów filmu
awardsRating	Ocena liczona na podstawie liczby nominacji oraz liczby zdobytych nagród
won	Liczba zdobytych nagród
nominations	Liczba nominacji filmu
duration	Czas trwania filmu
type	Lista gatunków filmowych
img	Ścieżka URL do plakatu filmowego
ratingFilmaster	Ocena w serwisie Filmaster.pl
ratingImdb	Ocena w serwisie Imdb.com
ratingFilmweb	Ocena w serwisie Filmweb.pl
release	Rok produkcji
originalTitle	Tytuł oryginalny
title	Tytuł w serwisie
__v	Automatyczny parametr wersji rekordu nadawany przez bazę danych

Tabela 5.2 Szczegóły encji films

Kolekcja users zawiera informacje o użytkownikach serwisu. Umożliwia im logowanie, rejestrację oraz autentykację, by mogli dostać się do zdefiniowanych zasobów dla danego typu użytkownika.

Atrybut	Opis
__id	Automatycznie generowany numer jednoznacznie identyfikujący użytkownika.
Salt	Parametr służący do kodowania i dekodowania hasła użytkownika do serwisu
displayName	Wyświetlana w serwisie nazwa użytkownika stworzona z konkatencji imienia i nazwiska użytkownika
Provider	Typ nadawany przez system rejestracji służący do identyfikowania środowiska
Username	Login do serwisu
Created	Data rejestracji użytkownika
Roles	Lista ról użytkownika
profileImageUrl	Relatywna ścieżka do avataru użytkownika
Password	Zaszyfrowane hasło użytkownika
Email	Email Użytkownika
lastName	Imię użytkownika
firstName	Nazwisko użytkownika
__v	Automatyczny parametr wersji rekordu nadawany przez bazę danych
resetPasswordExpires	Data wygaśnięcia Tokena do zmiany hasła
resetPasswordToken	Parametr dodawany przy resetowaniu hasła, umożliwia jego zmianę

Tabela 5.3 Szczegóły encji users

### 5.4.2 Model fizyczny

Model fizyczny w przypadku nierelacyjnej bazy danych MongoDB ogranicza się do utworzenia kolekcji:

Utworzenie kolekcji użytkowników:

```
db.createCollection(users)
```

Utworzenie kolekcji filmów:

```
db.createCollection(films)
```

Utworzenie kolekcji historii wyszukiwania:

```
db.createCollection(history)
```



## Rozdział 6

### Implementacja systemu do rekomendacji filmów

#### 6.1 Wprowadzenie

W tym rozdziale zostaną przedstawione kroki wykonane podczas implementacji serwisu zaczynając od wykorzystanych środowisk oraz narzędzi programistycznych. Następnym elementem będzie opis architektury wysokiego poziomu prezentujący interfejsy aplikacji oraz możliwe działania systemu. Kolejny podrozdział to krótkie zaprezentowanie najważniejszych kroków implementacyjnych, pokazanie ich możliwości oraz specyfikacja ich działania.

#### 6.2 Wykorzystane środowiska i narzędzia programistyczne

Pierwszym z narzędzi użytych jeszcze przed krokiem implementacji był Visual Paradigm 13.1. To w oparciu o niego powstały diagramy projektu aplikacji.

Wstępnym krokiem implementacyjnym było poznanie zalet jakie niesie za sobą narzędzie Yeoman. Jest to generator projektów zachowujący dobre praktyki programistyczne oraz pomagający w początkowej konfiguracji komponentów aplikacji. Wymaga on użycia szeregu elementów, które zostały przedstawione poniżej:

- Node Package Installer – narzędzie służące do instalacji pakietów wymaganych do wygenerowania projektu
- Node – serwerowe środowisko programistyczne zaprojektowane przy użyciu silnika JavaScript stworzonego przez Google.
- Bower – narzędzie służące do pobierania paczek klientów takich jak AngularJS dodawanych do projektu
- MongoDB – baza danych aplikacji
- Karma – narzędzie służące do testowania aplikacji
- Git – system kontroli wersji

Wspólne działanie tych komponentów pozwoliło na wygenerowanie projektu opartego na MongoDB, AngularJS, Express oraz Node.js. Jako środowisko programistyczne podczas realizacji tego projektu został użyty program JetBrains WebStorm 2016.1.1. Łączność z bazą danych zapewniał program Robomongo 0.9.0. Poniżej przedstawiono zalety i możliwości użycia tych programów:

- Większa swoboda w kompilowaniu i uruchomieniu kodu źródłowego z poziomu WebStorm w porównaniu z grunt.
- Wygodne GUI pozwalające na formatowanie wielu różnych języków programowania w WebStorm.
- Łatwość w sprawdzaniu stanu bazy danych, szczególnie w przypadku Web scrapingu. Program Robomongo udostępnia intuicyjne GUI pozwalające nawet początkującym użytkownikom na sprawdzenie stanu bazy danych, edytowanie rekordów, usunięcie ich. Wykonywanie tych operacji przypadku np. baz Oracle czy MySQL wymagałoby nauki specyficznego języka.

Do testowania aplikacji po stronie klienta służyły przeglądarki internetowe i dostępne w nich narzędzia programistyczne:

- Google Chrome wersja: 50.0
- Mozilla Firefox wersja: 46.0.1
- Internet Explorer wersja: 11

## 6.3 Architektura wysokiego poziomu

Architektura wysokiego poziomu aplikacji zostanie przedstawiona w formie zrzutów ekranu gotowej aplikacji wraz z możliwymi działaniami użytkownika.

### Logowanie użytkownika

Jest to ekran pojawiający się po przejściu do zakładki „Zaloguj się” w projektowanej aplikacji. Po poprawnym zalogowaniu użytkownik jest przekierowany na stronę główną aplikacji.

Logowanie

Login

admin

Hasło

.....

Zaloguj lub Zarejestruj

Zapomniałeś hasła?

Rysunek 6.1 Logowanie użytkownika – widok użytkownika

### Logowanie użytkownika – błąd logowania

Błąd logowania to komunikat pojawiający się w przypadku podania niepoprawnych danych użytkownika. Poniższy komunikat jest jednakowy dla wszystkich rodzajów błędów.

Logowanie

Login

admin

Hasło

.....

Zaloguj lub Zarejestruj

Zapomniałeś hasła?

Niepoprawny login lub hasło

Rysunek 6.1.1 Logowanie użytkownika (błąd logowania) – widok użytkownika

### Rejestracja użytkownika

Rejestracja użytkownika to widok pojawiający się po przejściu w zakładkę „Zarejestruj się” dostępną w menu głównym projektowanej aplikacji lub kliknięciu na link „Zarejestruj”

prezentowany na rysunku 6.1. Po poprawnej rejestracji użytkownik przekierowywany jest na stronę główną serwisu.

Rejestracja email

Imię

Nazwisko

Email

Login

Hasło

[Zarejestruj](#) lub [Zaloguj](#)

Rysunek 6.2 Rejestracja użytkownika – widok użytkownika

### Rejestracja użytkowników – walidacja –pól

Walidacja pól to ekran pojawiający się w przypadku wprowadzenia niepoprawnych danych podczas rejestracji. Wszystkie pola walidowane są pod względem wprowadzenia danych. Dodatkowo pojawia się pasek postępu ze względu na spełnienie wymagań dotyczących hasła.

Rejestracja email

Imię  
  
Imię wymagane

Nazwisko  
  
Nazwisko wymagane

Email  
  
Adres email niepoprawny

Login

Hasło


Wymagania hasła  
100%

[Zarejestruj](#) lub [Zaloguj](#)









Rysunek 6.2.2 Rejestracja użytkownika (walidacja danych) – widok użytkownika

### Lista filmów

Lista filmów to ekran pojawiający się po przejściu na zakładkę „Lista filmów” w menu głównym aplikacji. Pole powyżej listy jest odpowiedzialne za filtrowanie wyników, a pasek poniżej prezentowanych filmów jest odpowiedzialny za nawigację pomiędzy poszczególnymi elementami.

 Lista filmów Zarejestruj się Zaloguj się

Lista filmów

	<b>Jackie Brown</b>
	<b>Pulp Fiction</b> <small>Przemoc i odkupienie w opowieści o dwóch płatnych mordercach pracujących na zlecenie mafii, żonie gangstera, bokserze i parze okradającej ludzi w restauracji.</small>
	<b>Kill Bill</b> <small>Płatna zabójczyni zwana Czarną Mambą po czterech latach wybudza się ze śpiączki z zamiarem krwawej zemsty na ludziach, którzy próbowali ją zabić.</small>
	<b>Od zmierzchu do świtu</b> <small>Po wzięciu zakładników dwaj przestępcy uciekają do Meksyku. Zatrzymują się w przydrożnym lokalu, nie wiedząc, że to siedlisko wampirów.</small>
	<b>Kill Bill 2</b> <small>Czarna Mamba kontynuuje swoją krwawą zemstę na byłym szefie Billu i jego współpracownikach, którzy zniszczyli jej życie.</small>
	<b>Wściekle psy</b> <small>Po nieudanym napadzie na sklep jubilerski kilku złodziei zdaje sobie sprawę, że jest wśród nich zdrajca, który powiedział policji o ich planie.</small>
	<b>Nienawistna ósemka</b> <small>Dwaj łowcy głów, próbując znaleźć schronienie przed zamiecią śnieżną, trafiają do Wyoming, gdzie wplątani zostają w spłot krwawych wydarzeń.</small>
	<b>Clerks - Sprzedawcy</b> <small>Małe sklepiki, takie jak w filmie "Clerks - Sprzedawcy", nie cieszą się przeważnie najlepszą renomą i często są tematem kpin. Bo i towar nie taki, ceny wyższe, a termin przydatności do spożycia upływa za kilka dni. I jeśli sprzedawcy uda się dobrze wydać resztę, klienci są zadowoleni, niczego więcej nie oczekując.</small>

First Previous 1 Next Last

Rysunek 6.3 Lista filmów – widok użytkownika

## Szczegóły użytkownika

Szczegóły użytkownika to widok pojawiający się po przejściu w zakładkę „Zarządzanie użytkownikami” i wybranie konkretnego użytkownika z listy. Ikony na środku prezentowanego ekranu odpowiedzialne są odpowiednio za edycję danych i usunięcie użytkownika.

Panel administracyjny ▾

Lista filmów

Scrapper

★

m m ▾

mat

✎

🗑

First Name	Mateusz
Last Name	Brożyna
Email	matbrozyna7@gmail.com
Provider	local
Created	2016-04-14T20:21:02.432Z
Roles	user

Rysunek 6.4 Szczegóły użytkownika – widok użytkownika

## Wyszukiwarka filmów

Wyszukiwarka filmów to ekran główny aplikacji udostępniający wyszukiwarke filmową. Uruchomienie wyszukiwania powoduje przekierowanie do widoku szczegółów filmu lub pojawienie się komunikatu o braku wyników.

Lista filmów

Zarejestruj się

Zaloguj się

Wyszukiwarka filmów

z serwisów Imdb.com, Filmweb.pl oraz Filmaster.pl

Kraj

USA

Wielka Brytania

Francja

Polska

Maksymalna długość

Czas trwania w minutach

Reżyser

Reżyser

Typ

Przygodowy

Dramat

Romans

Dramat obyczajowy

Aktorzy

Aktorzy

Strona źródłowa

☐ imdb.com
 ☐ filmweb.pl
 ☐ filmaster.pl

Data od

Data do

Rok produkcji w formacie yyyy

Rok produkcji w formacie yyyy

Ocena

Przykład: 9,8

☐ Wygrane

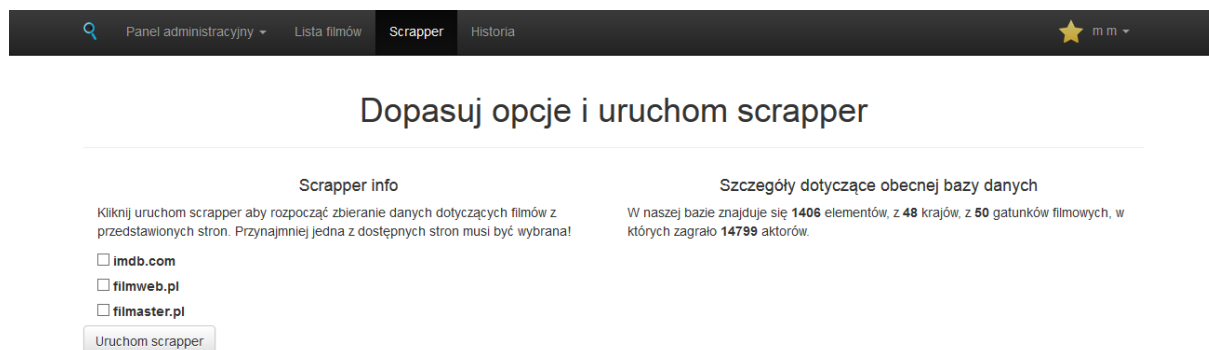
☐ Sumarycznie

Uruchom wyszukiwanie

Rysunek 6.5 Wyszukiwarka – widok użytkownika

## Scrapper

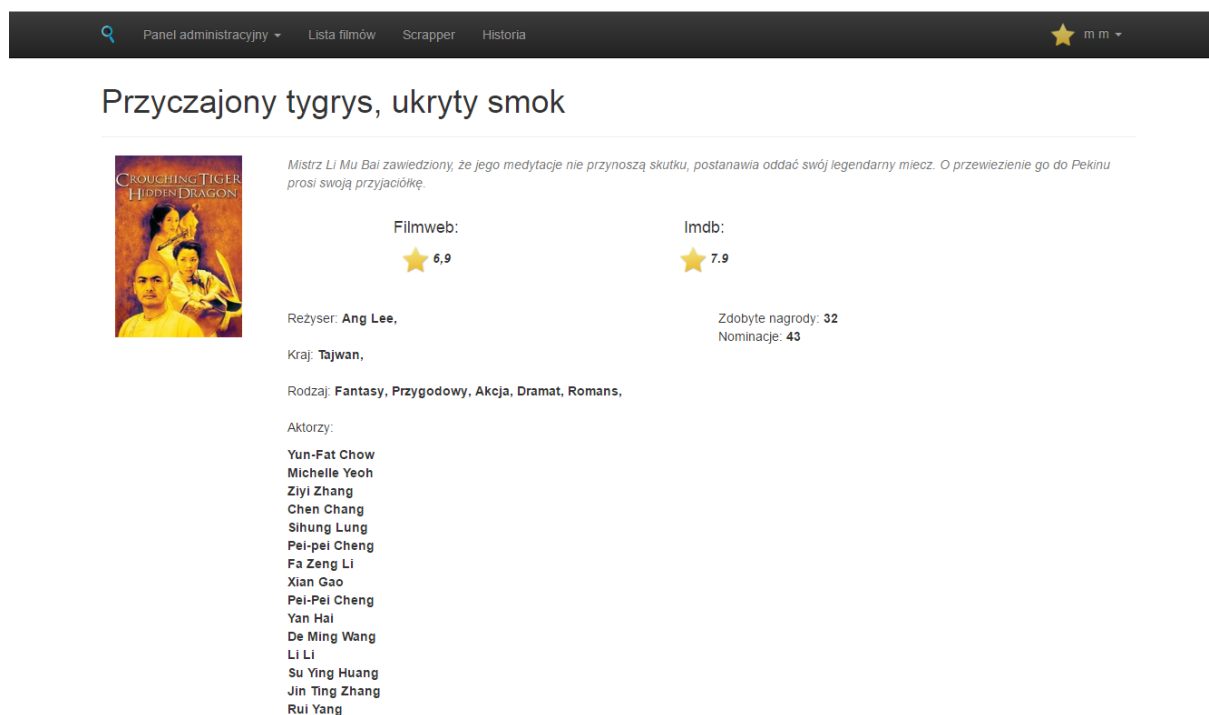
Scrapper to ekran pojawiający się po przejściu w zakładkę „Scrapper” tworzonej aplikacji. Jest to widok dostępny dla użytkowników z prawami administratora, pozwalający na uruchomienie Web scraper’a do zbierania danych. Dodatkowo zawiera szczegółowe informacje dotyczące bazy danych.



Rysunek 6.6 Scrapper – widok administratora

## Szczegóły filmu

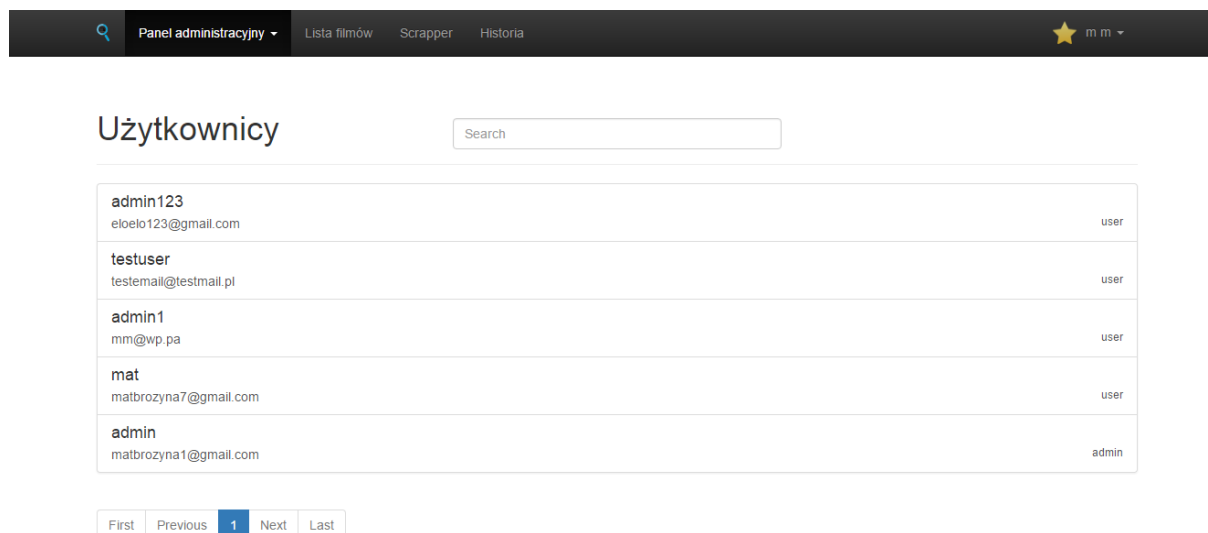
Szczegóły filmu to widok pojawiający się po wybraniu filmu z listy zaprezentowanej na rysunku 6.3.



Rysunek 6.7 Szczegóły filmu – widok użytkownika

## Zarządzanie użytkownikami

Jest to ekran aplikacji pojawiający się po przejściu do zakładki zarządzanie użytkownikami dostępnej w panelu administracyjnym. Widok jest dostępny dla użytkowników z rolą administratora. Wybranie użytkownika z listy powoduje przejście do widoku szczegółów użytkownika prezentowanego na rysunku 6.4.



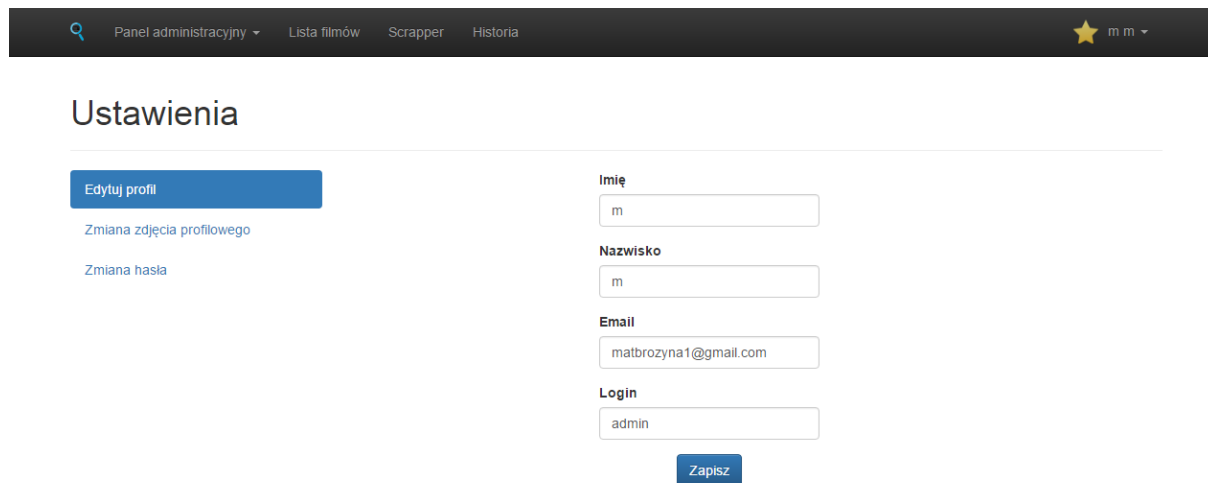
Użytkownicy		Search
admin123	eloelo123@gmail.com	user
testuser	testemail@testmail.pl	user
admin1	mm@wp.pa	user
mat	matbrozyna7@gmail.com	user
admin	matbrozyna1@gmail.com	admin

First Previous 1 Next Last

Rysunek 6.8 Lista użytkowników– widok administratora

## Ustawienia konta – widok 1

Widok ustawienia konta pojawia się po przejściu do zakładki „Edytuj profil” dostępnej w menu użytkownika po prawej stronie panelu. Umożliwia zmianę danych zapisanych w bazie danych.



Ustawienia

Edytuj profil

Zmiana zdjęcia profilowego

Zmiana hasła

Imię

m

Nazwisko

m

Email

matbrozyna1@gmail.com

Login

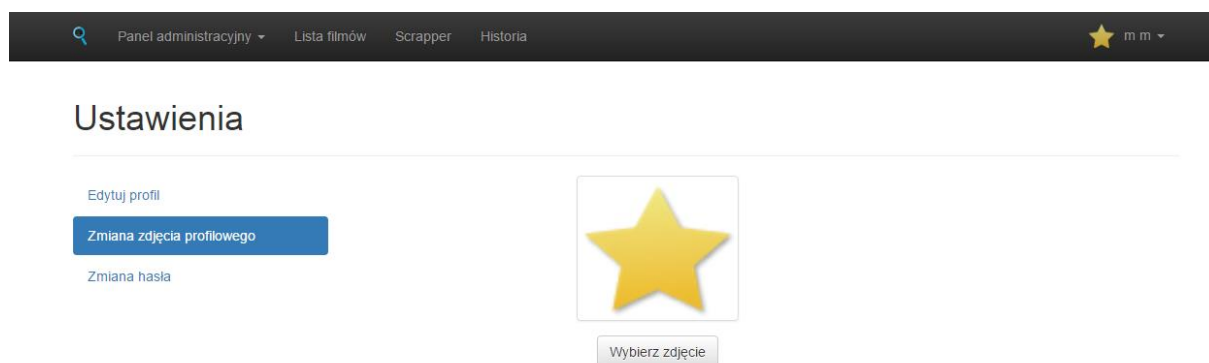
admin

Zapisz

Rysunek 6.9 Ustawienia profilu (edycja danych) – widok użytkownika

## Ustawienia konta – widok 2

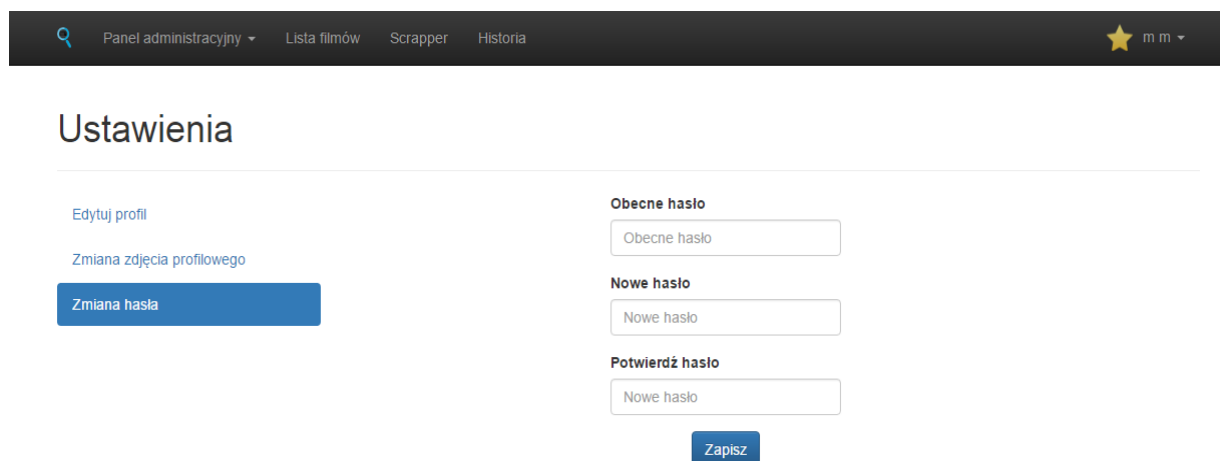
Widok pojawia się po przejściu do zakładki „Zmiana zdjęcia profilowego” dostępnego w wysuwanym panelu użytkownika lub w menu widocznym na rysunku 6.9. Umożliwia zmianę avataru widocznego w pasku menu.



Rysunek 6.10 Ustawienia profilu (zmiana zdjęcia profilowego) – widok użytkownika

### Ustawienia konta – widok 3

Widok pojawia się po wybraniu zakładki „Zmiana hasła” widocznej w menu użytkownika lub w ustawieniach profilu. W przypadku błędnie wprowadzonych danych pojawia się błąd walidacji.

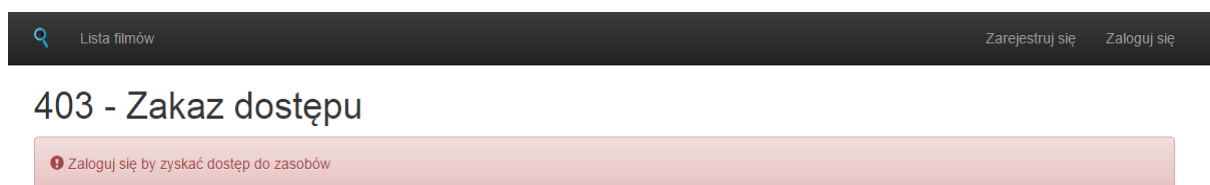


Rysunek 6.11 Ustawienia profilu (zmiana hasła) – widok użytkownika

### 403 Forbidden

Ekran zakaz dostępu pojawia się w przypadku braku uprawnień do zasobów.

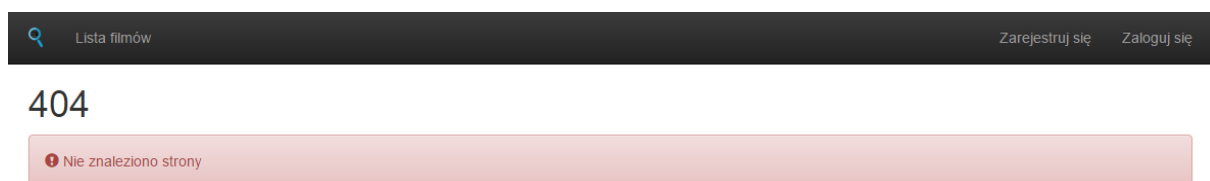




Rysunek 6.12 Zakaz dostępu

## 404 Not found

Ekran pojawiający się w przypadku żądania nieistniejących zasobów.



Rysunek 6.13 Nie znaleziono strony

## Historia wyszukiwania – widok 1

Widok historia wyszukiwania dostępny jest po wybraniu zakładki „Historia” z menu głównego. Jest dostępny dla zalogowanych użytkowników. Udostępnia listę wyszukiwania użytkownika z datą wyszukiwania. Wybranie elementu powoduje przejście do ekranu „Historia wyszukiwania – szczegóły”.

## Historia wyszukiwania

Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 2, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna
Znaleziony Jun 1, 2016 przez Mateusz Brożyna

Rysunek 6.14 Historia wyszukiwania – widok użytkownika

### Historia wyszukiwania – widok 2

Widok w przypadku braku historii wyszukiwania użytkownika. Widok zmienia się na listę przedstawioną 6.14 w przypadku pozytywnego wyszukiwania filmu.

## Historia wyszukiwania

Brak historii

Rysunek 6.15 Historia wyszukiwania (brak historii) – widok użytkownika

### Historia wyszukiwania – szczegóły

Ekran szczegóły wyszukiwania pojawia się po wybraniu elementu z listy przedstawionej w zakładce „Historia”. Przedstawia znaleziony za pomocą wyszukiwarki film.

## Skazani na Shawshank



Adaptacja opowiadania Stephena Kinga. Historia niesłusznie skazanego na dożywocie bankiera, który musi przeżyć w brutalnym świecie rządzonego przez strażników i współwięźniów.

Filmweb:



Imdb:



Reżyser: **Frank Darabont,**

Zdobyte nagrody: **6**

Nominacje: **19**

Kraj: **USA,**

Rodzaj: **Dramat, Gangsterski,**

Aktorzy:

**Tim Robbins**  
**Morgan Freeman**  
**Bob Gunton**  
**William Sadler**  
**Clancy Brown**  
**Gil Bellows**  
**Mark Rolston**

Rysunek 6.16 Szczegóły wyszukiwania – widok użytkownika

## 6.4 Wyszukiwarka filmów

Pierwszym elementem, który zostanie przedstawiony w prezentacji wyszukiwarki filmów są kryteria wyszukiwania oraz zasada ich działania w zapytaniach do bazy danych. Użytkownik ma możliwość wybrania listy krajów z których pochodzi film, reżysera, aktorów, przedział daty produkcji filmu, maksymalną długość, listę typów filmu, strony źródłowe, minimalną ocenę w serwisie oraz kategorię wyszukiwania. Wyniki są filtrowane na podstawie przedstawionych poniżej atrybutów w następujący sposób:

- Lista krajów - filtruje filmy ze względu na kraj powstania – co najmniej jeden kraj powstania znajdujący się w bazie danych danego filmu musi odpowiadać krajom wskazanym przez użytkownika. Jeśli lista krajów nie jest oznaczona, wynikowa lista filmów nie jest filtrowana po tej wartości.
- Przedział daty produkcji filmu – filtruje filmy ze względu na datę produkcji. Zakres dat obejmuje przedziały obustronnie domknięte, co oznacza, że w przypadku wprowadzenia parametru „data od” szukane są filmy równe lub starsze od wprowadzonej wartości, a w przypadku wprowadzenia wartości „data do” szukane są filmy równe lub młodsze od wprowadzonej wartości. Parametry dat łączone są za pomocą operatora logicznego AND. W przypadku gdy obie wartości nie są oznaczone, wynikowa lista filmów nie jest filtrowana po tej wartości.
- Lista typów filmów – filtruje filmy ze względu na typ filmu – co najmniej jeden typ filmu znajdujący się w bazie danych musi odpowiadać typom wskazany

przez użytkownika. Jeśli lista typów nie jest oznaczona, wynikowa lista filmów nie jest filtrowana po tej wartości.

- Strony źródłowe – użytkownik podaje serwisy źródłowe na podstawie których ma znaleźć wyniki wyszukiwania. W przypadku, gdy żadna strona źródłowa nie została podana wyszukiwarka uruchamia się, a wynikowa lista zawiera filmy ze wszystkich serwisów źródłowych. W przypadku podania więcej niż jednego źródła system wyszukuje film z dowolnego wprowadzonego serwisu.
- Minimalna ocena – w przypadku gdy użytkownik wprowadzi minimalną ocenę filmu system uruchamia zapytania filtrując wyniki, dla zadanych stron źródłowych w których:

$$ocena > minimalnaOcena$$

*ocena* – wartość z poszczególnych stron źródłowych znajdująca się w bazie danych.

W przypadku gdy parametr jest pomijany system szuka elementów dla danych stron źródłowych z kryterium:

$$ocena > 0$$

Wartość parametru jest pomijana, gdy nie wprowadzono żadnej strony źródłowej. Gdy wprowadzono więcej niż jedną stronę źródłową filtr akceptuje dany film zgodnie z nierównością:

$$\begin{array}{l} ocenaImdb > minimalnaOcena \\ ocenaFilmweb > minimalnaOcena \\ ocenaFilmaster > minimalnaOcena \end{array}$$

Wszystkie przedstawione powyżej parametry filtrowania są łączone ze sobą za pomocą operatora logicznego AND.

Dodatkowo wyszukiwarka zawiera następujące pola używane w kategorii *Ocena sumaryczna*:

- Reżyser – lista szukanych reżyserów
- Aktorzy – lista szukanych aktorów
- Maksymalna długość – maksymalna długość filmu.

Jeśli chodzi o realizację algorytmu wyszukiwania to był on podzielony na kategorie według których użytkownik mógł znaleźć film:

- *Wygrane*
- *Ocena sumaryczna*
- *Brak kryterium*

Wygrane to ocena bazująca na nominacjach do nagród oraz wygranych w konkursach filmowych. Na podstawie zebranych danych dotyczących danego filmu i zdobytych przez niego nagród obliczana była ocena sumaryczna – przedstawiana w bazie danych jako pole *awardsRating* wyznaczone według wzoru:

$$awardsRating = nominations + 5 * won$$

*Nominations* – liczba nominacji przedstawiona na stronach źródłowych  
*won* – liczba zwycięstw.

W przypadku braku wyboru opcji kryterium system wybierał pierwszy film spełniający podane przez użytkownika kryteria wyszukiwania zgodnie z opisanym wcześniej algorytmem filtrowania.

Ostatnim przypadkiem jest *Ocena sumaryczna*. Jest to algorytm generujący wynikową listę filmów na podstawie wprowadzonych parametrów filtrowania. Następnie dla każdego filmu na podstawie przedstawionych parametrów niefiltrujących oraz oceny w poszczególnych serwisach obliczana jest ocena sumaryczna zgodnie z wzorem:

$$\begin{aligned} &Ocenasumaryczna \\ &= liczbaZnalezionychAktorów * 10 + liczbaZnalezionychReżyserów \\ &\quad * 10 + długość + ocena \end{aligned}$$

*Długość* – to stała wartość 5 – jeśli film jest krótszy niż maksymalny wprowadzony czas  
*LiczbaZnalezionychAktorów* – to liczba aktorów znajdujących się w bazie filmu, wprowadzonych przez użytkownika w polu Aktorzy  
*LiczbaZnalezionychReżyserów* – to liczba reżyserów znajdujących się w bazie filmu, wprowadzonych przez użytkownika w polu Reżyser  
*Ocena* – średnia ocen w bazie danych ze wszystkich źródeł

## 6.5 Web Scraper

Na początku kilka słów wyjaśnienia odnośnie Web scrapingu. Jest to technika internetowa pozwalająca na zbieranie danych ze stron internetowych. W tym przypadku polegała ona na znalezieniu podstron serwisu źródłowego, a następnie wyciągnięciu informacji o filmie na podstawie posiadanego kodu źródłowego strony. W związku z liczbą stron źródłowych wymagane było zaimplementowanie 3 takich robotów – jeden dla każdego źródła. W tym momencie warto opisać sposób implementacji tych robotów z wyszczególnieniem elementów specyficznych dla danego serwisu:

- Filmweb.pl – Scrapper zbiera wszystkie znajdujące się w bazie informacje dotyczące filmu. Na podstawie zakładki „baza filmów” generowane są po stronie robota linki w następujący sposób:

<http://www.filmweb.pl/search/film?q=&type=&startYear=&endYear=&countryIds=&genreIds=&startRate=&endRate=&startCount=&endCount=&sort=COUNT&sortAscending=false&c=portal&page=1>

Wartość page zmieniana jest w zależności od liczby oczekiwanych wyników.

Z każdej tak wygenerowanej strony system zbiera linki do 10 filmów przedstawionych na stronie i zapisuje je do wynikowej tabeli. Po zdobyciu wszystkich oczekiwanych adresów system przechodzi do właściwego zbierania danych. Samo gromadzenie rekordów polega najpierw na zdobyciu wszelkich możliwych informacji z URL'a źródłowego a następnie przejściu do zakładki /awards i zebraniu danych dotyczących nagród zdobytych przez dany film. Tak zebrane informacje zostają dodane do systemu. Baza danych filmweb jest głównym źródłem posiadanych informacji w serwisie, a co za tym idzie w tym przypadku nie jest wymagane przeprowadzenie procesu ETL po stronie Scrappera.

- Imdb.com – Scraper zbiera wszystkie informacje znajdujące się na stronie źródłowej poza opisem filmu (pomijane, ponieważ nawet w polskiej wersji serwisu opisy są w języku angielskim) oraz nagrodami (pominięte, ponieważ w niektórych sytuacjach prezentowane są w sposób nie pozwalający jednoznacznie odróżnić zdobytych nagród od nominacji). Samo zbieranie danych polega na wygenerowaniu adresów URL na podstawie rankingu top 500, a następnie przejściu po wszystkich wygenerowanych linkach i zgromadzeniu informacji.
- Filmaster.pl – Scraper zbiera informacje dotyczące tytuły filmu, linka do plakatu filmowego, rok produkcji, ocenę serwisu, opis filmu oraz aktorów. Na podstawie zakładki „rankingi” generowane są po stronie robota linki w następujący sposób:

<http://filmaster.pl/rankingi/?page=2>

Wartość page zmieniana jest w zależności od liczby oczekiwanych wyników. Z każdej tak wygenerowanej strony system zbiera linki do 10 filmów przedstawionych na stronie i zapisuje je do wynikowej tabeli. Po zdobyciu wszystkich oczekiwanych adresów system przechodzi do właściwego zbierania danych.

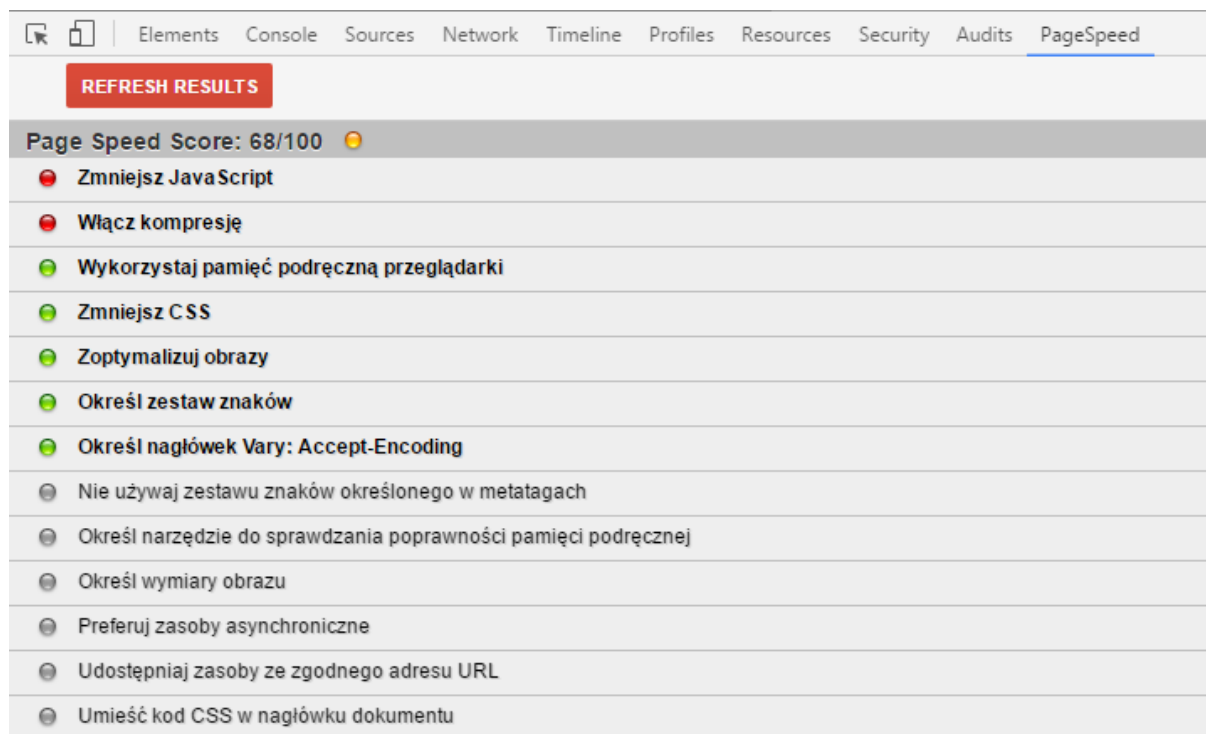
Najtrudniejszym elementem podczas implementacji była konieczność połączenia wyników w jeden spójny model bazy danych. W związku z użyciem 3 różnych serwisów, których modele różnią się, w przypadku dwóch z nich wymagane było przeprowadzenie procesu ETL po stronie systemu. Proces ten w przypadku Imdb.com polegał na mapowaniu kraju oraz gatunku filmowego. Dodatkowym aspektem wymagającym implementacji była konieczność sprawdzania wszystkich możliwych konfiguracji danego tytułu by uniknąć redundancji danych. Proces ten udało się zoptymalizować porównując oryginalne tytuły filmów (Brak możliwości porównania tytułów polskojęzycznych z powodu braku polskich znaków w serwisie Imdb).

## Rozdział 7

### Testy

#### 7.1 Testy poprawności działania

Testy poprawności działania aplikacji zostały przeprowadzone za pomocą prezentowanych w podrozdziale 6.2 przeglądarek internetowych. Dodatkowo za pomocą wtyczki Page Speed Insights przeprowadzono testy szybkości działania aplikacji:



Rysunek 7.1 Wynik badania szybkości działania aplikacji

Biorąc pod uwagę liczbę zasobów, wynik można uznać za zadowalający. W przyszłości warto pomyśleć o kompresji paczek klienckich, co pozwoli na otrzymanie lepszych rezultatów.

#### 7.2 Testy jednostkowe i testy bezpieczeństwa

Z powodu stale zmieniającej się bazy danych niemożliwe było napisanie odpowiednich testów wyszukiwarki, dających jednoznaczne wyniki. Jeśli chodzi o mechanizm zbierania danych utworzone testy zostały napisane tak, by badały liczbę elementów w poszczególnych krokach metody.

Bezpieczeństwo aplikacji zostało przetestowane pod względem dostępu do danych w zależności od roli użytkownika. Badane przypadki nie wykazały niepożądanych wycieków. Przykład testu:

---

### Test dostępu do zasobów

---

```
it('should be able to get a list of Histories if not signed in', function (done) {
  var historyObj = new History(history);
  // Save the history
  historyObj.save(function () {
    // Request Histories
    request(app).get('/api/histories').end(function (req, res) {
      res.body.should.be.instanceof(Array).and.have.lengthOf(1);
      done();
    });
  });
});
```

---

### 7.3 Podsumowanie

Implementowana aplikacja przechowuje dane, które mogą zostać w każdej chwili odtworzone w szybkim czasie. Główny nacisk został położony na testy poprawności działania oraz jednostkowe testy użycia Scrapera. Wszystkie przeprowadzone testy spełniają założenia projektowe, oraz нефункционалне założenia dotyczące przechowywanych haseł i autoryzacji użytkowników.



## Rozdział 8

### Podsumowanie pracy

Prace wykonane podczas projektu można podzielić na dwie kategorie. Pierwsza z nich – opracowanie projektu aplikacji podzielona była na kilka etapów takich jak:

- Opracowanie wymagań funkcjonalnych i нефункциональных – sformułowane w taki sposób by aplikacja mogła konkurować z innymi serwisami internetowymi oraz dodatkowo wprowadzała nowe funkcjonalności
- Opisanie przypadków użycia – przygotowane na podstawie wcześniej przedstawionych wymagań. Przypadki użycia oparte były na diagramie przypadków użycia, diagramie aktywności użytkownika oraz scenariuszach.
- Utworzenie interfejsu aplikacji – wykonany w formie diagramu przepływu
- Baza danych – zamodelowana na podstawie wymagań funkcjonalnych i нефункциональных oraz innych serwisów bazowych.

Drugim etapem były prace związane z przygotowaniem prototypu aplikacji na podstawie wyżej wymienionych wymagań. Implementacja w tym zakresie rozdzielona była na trzy etapy:

- Utworzenie szkieletu strony z logowaniem, rejestracją i stroną główną aplikacji
- Opracowanie robota internetowego do zbierania danych filmowych z dostępnych w sieci baz filmowych
- Implementacja wyszukiwarki filmowej ze względu na podane przez użytkownika kryteria wyszukiwania

Utworzony prototyp aplikacji realizuje wszystkie założone wcześniej funkcjonalności, a całość projektu jest odzwierciedleniem tematu pracy i przedstawionych w niej celów.

Najtrudniejszym elementem podczas realizacji tego projektu było zebranie danych służących do dalszej implementacji. Każde źródło z którego korzysta aplikacja ma inną koncepcję bazy danych, co po stronie systemu wymagało szeregu procesów ich ujednolicenia. Efekt końcowy jest zadowalający, jednak w niektórych przypadkach wyeliminowanie problemów z tym związanych było niemożliwe np. z powodu błędów w stronach źródłowych. Działanie mechanizmu wyszukiwania daje dobre wyniki, które z pewnością pozwolą trafić w gusta użytkowników. Pierwszą sugerowaną zmianą, która dotyczy samej wyszukiwarki byłoby zwiększenie bazy filmów. W wielu przypadkach pomimo poprawnego i założonego działania mechanizmu, wyszukiwarka zwracała wyniki odbiegające od oczekiwań z powodu niepełnych informacji lub ich braku. Bardzo pozytywnym aspektem była możliwość poznania technologii MEAN.JS. Pomimo braku wcześniejszych doświadczeń z programowaniem w JavaScript prace nad aplikacją okazały się nieskomplikowane, a wszelkie działania związane z bazą danych jasne i bardzo intuicyjne.

Analiza serwisów na podstawie których aplikacja została zaprojektowana i zaimplementowana to źródła, na podstawie których powinniśmy czerpać główną wiedzę dotyczącą nowych funkcjonalności. By aplikacja mogła na równi konkurować z tymi stronami wymagane jest jeszcze wiele zmian. Najciekawszym z rozwiązań byłoby utworzenie drugiego systemu rekomendacji – na podstawie znalezionych filmów i kryteriów wyszukiwania.

## Bibliografia

- [1] Google, *AngularJS by Google*, <http://www.angularjs.org>, [last access: 25.05.2016].
- [2] Ilhami M., Suharjito, *Film recommendation systems using matrix factorization and collbaorative filtering*, Information Technology Systems and Innovation (ICITSI), 2014 International Conference on. IEEE, 2014.
- [3] Oficjalna strona projektu MEAN.JS, <http://www.meanjs.org/>, [last access: 25.05.2016].
- [4] Sarwar B., Karypis G., Konstan J., Riedl J., *Item based Collaborative Filtering Recommendation Algorithms*, Proceedings of the 10th international conference on World Wide Web. ACM, 2001.
- [5] Su X., Khoshgoftaar TM., *A Survey of Collaborative Filtering Techniques*, Advances in artificial intelligence, 2009, 4
- [6] Witryna projektu Filmaster, <http://www.filmaster.pl/>, [last access: 25.05.2016].
- [7] Witryna projektu Filmweb, <http://www.filmweb.pl/>, [last access: 25.05.2016].
- [8] Witryna projektu SuggestMeMovie, <http://www.suggestmemovie.com/>, [last access: 25.05.2016].
- [9] Witryna projektu „The Internet Movie Database”, <http://www.imdb.com/>, [last access: 25.05.2016].
- [10] Yeoman©, *What's Yeoman?*, <http://www.yeoman.io>, [last access: 25.05.2016].
- [11] Zhao, Zhi-Dan, Ming-Sheng Shang, *User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop*, Knowledge Discovery and Data Mining, 2010, WKDD'10. Third International Conference on. IEEE, 2010.