

POLITECHNIKA WROCŁAWSKA  
KATEDRA INFORMATYKI TECHNICZNEJ

BAZY DANYCH 2

# Domowa wypożyczalnia wideo

*Magdalena Biernat*

*Mateusz Bortkiewicz*

Prowadzący  
dr inż. Tomasz Janiczek

21 stycznia 2018

# Spis treści

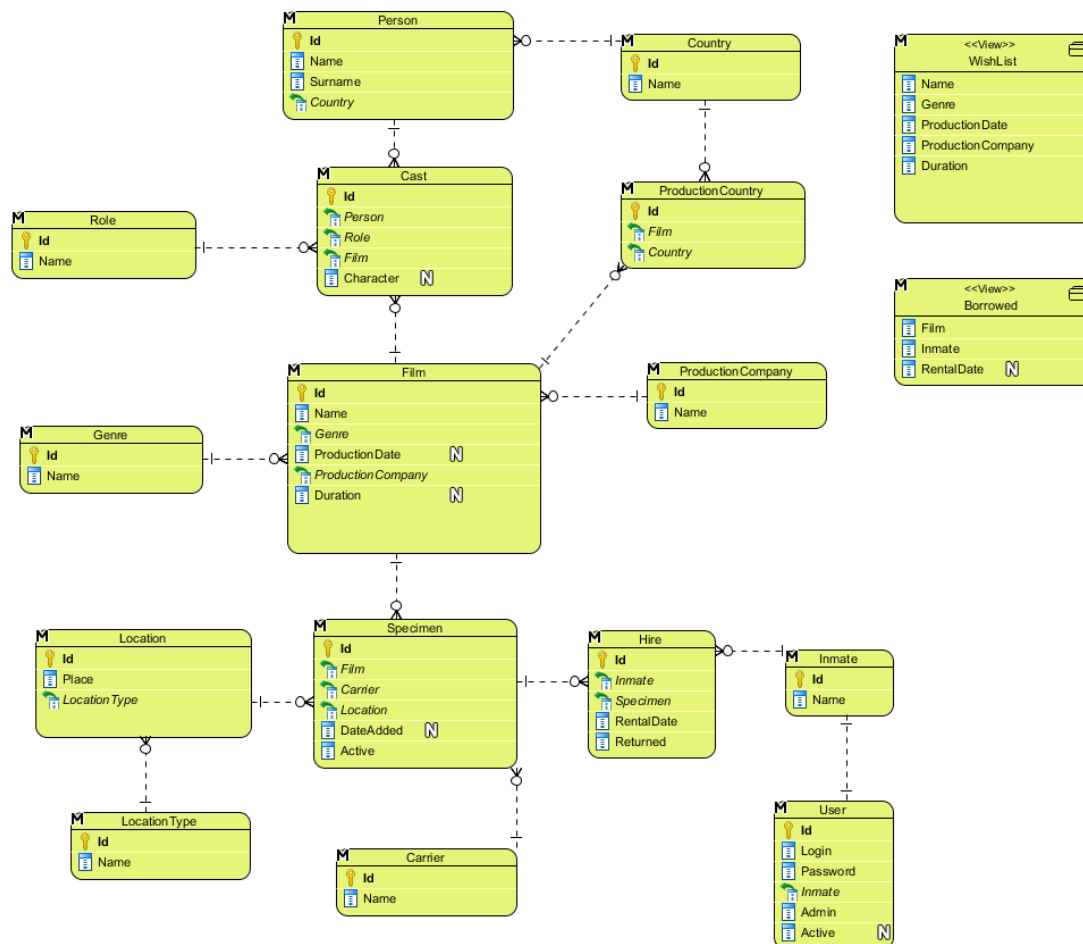
<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
<b>2</b>	<b>Model konceptualny bazy danych</b>	<b>2</b>
<b>3</b>	<b>Opis "świata rzeczywistego"</b>	<b>3</b>
3.1	Opis zasobów ludzkich . . . . .	3
3.2	Przepisy . . . . .	3
3.3	Dane techniczne . . . . .	3
<b>4</b>	<b>Wymagania funkcjonalne i нефункционалне</b>	<b>4</b>
4.1	Wymagania funkcjonalne . . . . .	4
4.2	Wymagania нефункционалне . . . . .	4
<b>5</b>	<b>Diagram przypadków użycia</b>	<b>5</b>
<b>6</b>	<b>Diagram związków encji</b>	<b>6</b>
<b>7</b>	<b>Analiza ilości encji</b>	<b>7</b>
7.1	Analiza liczby instancji dla każdej encji . . . . .	7
7.2	Analiza użycia dla każdej encji . . . . .	7
<b>8</b>	<b>Implementacja bazy danych</b>	<b>8</b>
8.1	Serwer bazodanowy . . . . .	8
8.2	Tworzenie tabel i wprowadzanie danych . . . . .	9
<b>9</b>	<b>Polityka bezpieczeństwa</b>	<b>9</b>
9.1	Zabezpieczenie serwera . . . . .	9
9.2	Dostęp do kont użytkowników . . . . .	9
9.3	Sniffing . . . . .	9
<b>10</b>	<b>Aplikacja desktopowa</b>	<b>11</b>
10.1	Środowisko i wykonanie . . . . .	11
10.2	Struktura projektu . . . . .	11
10.3	Połączenie z bazą danych . . . . .	12
10.4	Logowanie do aplikacji . . . . .	12
10.5	Okno główne . . . . .	14
10.5.1	Panel użytkownika . . . . .	14
10.5.2	Panel administratora . . . . .	15
10.6	Funkcjonalności . . . . .	15
10.7	Tytuły filmów . . . . .	16
<b>11</b>	<b>Testy aplikacji</b>	<b>17</b>
<b>12</b>	<b>Podsumowanie</b>	<b>17</b>

# 1 Wprowadzenie

Niniejszy dokument powstał z myślą o projekcie na Bazy Danych 2. Jest to omówienie poszczególnych zagadnień jak i funkcjonalności stworzonej na potrzeby projektu aplikacji.

## 2 Model konceptualny bazy danych

Poniżej przedstawiono model konceptualny naszej bazy. Zależało nam na tym, aby w bazie danych tabel było wystarczająco, tzn. była zaprezentowana każda relacja (jeden do jednego, jeden do wielu, wiele do wielu) oraz program implementował podstawowe funkcjonalności logowania i biblioteki. W tym celu posiadamy table skoncentrowane na samej bibliotece (Film, Specimen, Location etc), na użytkowniku (User, Inmate), oraz tabelę asocjacyjną wypożyczeń (Hire).



Rysunek 1: Model konceptualny

### 3 Opis "świata rzeczywistego"

Opis "świata rzeczywistego" - aplikacja "Domowa wypożyczalnia wideo"

#### 3.1 Opis zasobów ludzkich

- Głowa rodziny, tudzież osoba wyznaczona w domu administruje i zarządza aplikacją desktopową "Domowa wypożyczalnia wideo". Może ona usuwać i dodawać użytkowników, resetować hasła, zwracać/wypożyczać administracyjnie egzemplarze filmów, usuwać tytuły filmów, wycofywać z użycia egzemplarze filmów, a także wykonywać to co zwykły użytkownik
- Użytkownik, mieszkaniec domostwa może dodać lub edytować tytuły filmów oraz egzemplarze do tytułów obecnych już filmów. Może wypożyczać dostępny egzemplarz filmu
- Wypożyczenie posiada identyfikator domownika, identyfikator egzemplarza filmu a także datę wypożyczenia i informację czy film jest zwrócony.
- Użytkownik może edytować swoje konto, tj. zmieniać hasło.

#### 3.2 Przepisy

Liczba wypożyczonych jednocześnie egzemplarzy przez danego użytkownika nie może być większa niż trzy. Nie istnieje termin zwrotu - wypożyczenie jest bezterminowe. Nieaktywny użytkownik nie może się zalogować na konto i wypożyczyć/zwrócić filmu.

#### 3.3 Dane techniczne

Obsługa konta, bazy wideo i wypożyczeń powinna być dostępna przez aplikację desktopową połączoną z bazą danych. Wypożyczający loguje się do konta za pomocą loginu i hasła. Hasło nie jest przechowywane w bazie danych (jest to jedynie zasolone i haszowanie hasło).

## 4 Wymagania funkcjonalne i нефункционалне

### 4.1 Wymagania funkcjonalne

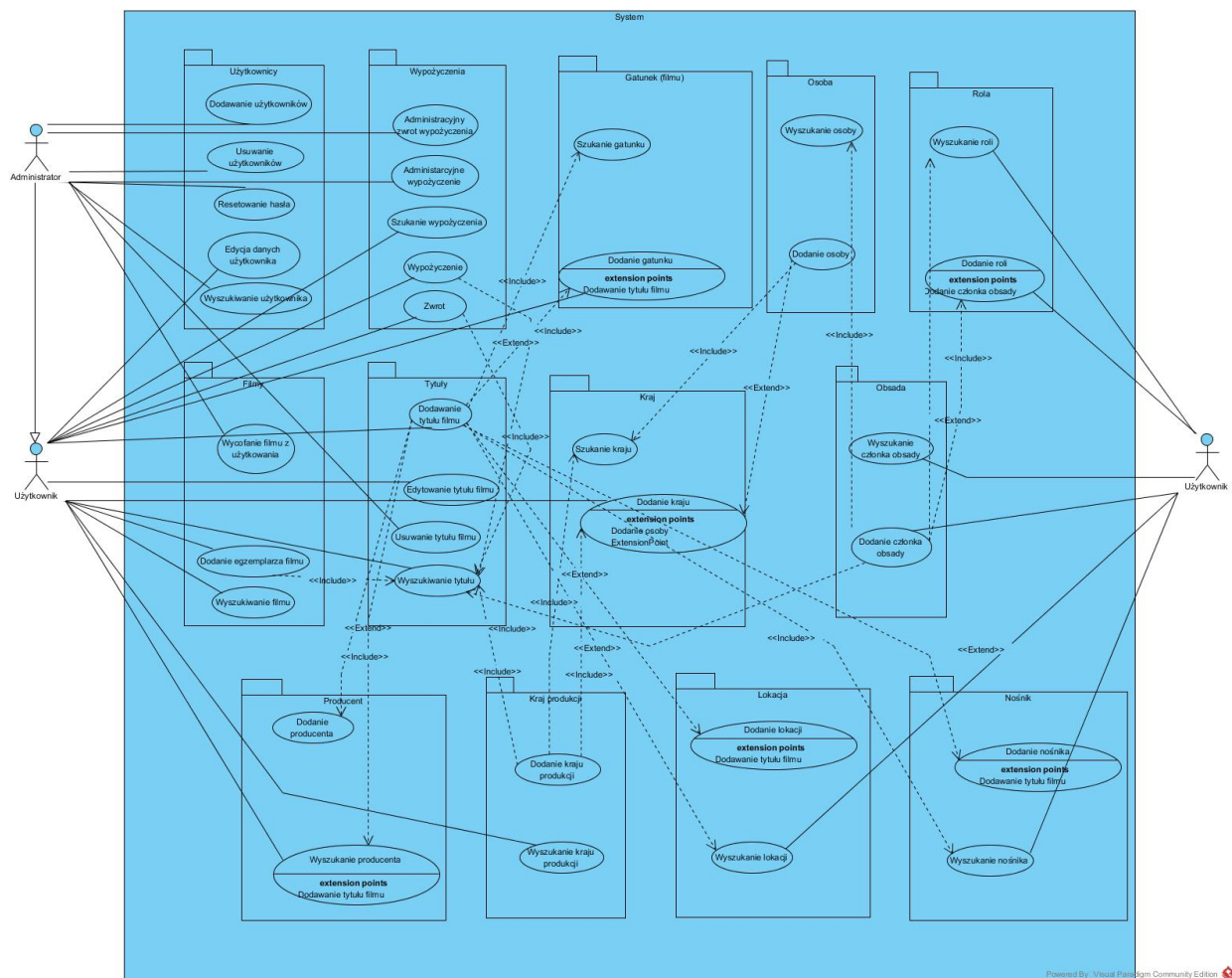
- aplikacja ma mieć możliwość wyszukiwania\дodawania\edytowania\usuwania filmów do bazy danych
  - wyszukiwać i dodawać filmy może każdy użytkownik
  - usuwać i edytować może tylko administrator systemu
- aplikacja ma mieć możliwość edytowania danych użytkownika
- dodawać nowego użytkownika może tylko administrator
- aplikacja ma mieć możliwość wyszukiwania\дodawania\edytowania\usuwania tytułów filmów do bazy danych
  - wyszukiwać i dodawać tytuły może każdy użytkownik
  - usuwać i edytować może tylko administrator systemu
- obsadę może dodawać każdy użytkownik
- aplikacja pokazuje wypożyczone pozycje i listę życzeń (filmy, których egzemplarze nie są umieszczone w bazie danych)

### 4.2 Wymagania нефункционалне

- ilość domowników nie wpływa na szybkość działania systemu
- ilość filmów\tytułów\уżytkowników etc. Jest ograniczona tylko pojemnością dysku na którym stoi baza danych
- przyjazny interfejs użytkownika
- liczba błędów w aplikacji przez pierwszy miesiąc od wydania aplikacji nie może przekroczyć 5

## 5 Diagram przypadków użycia

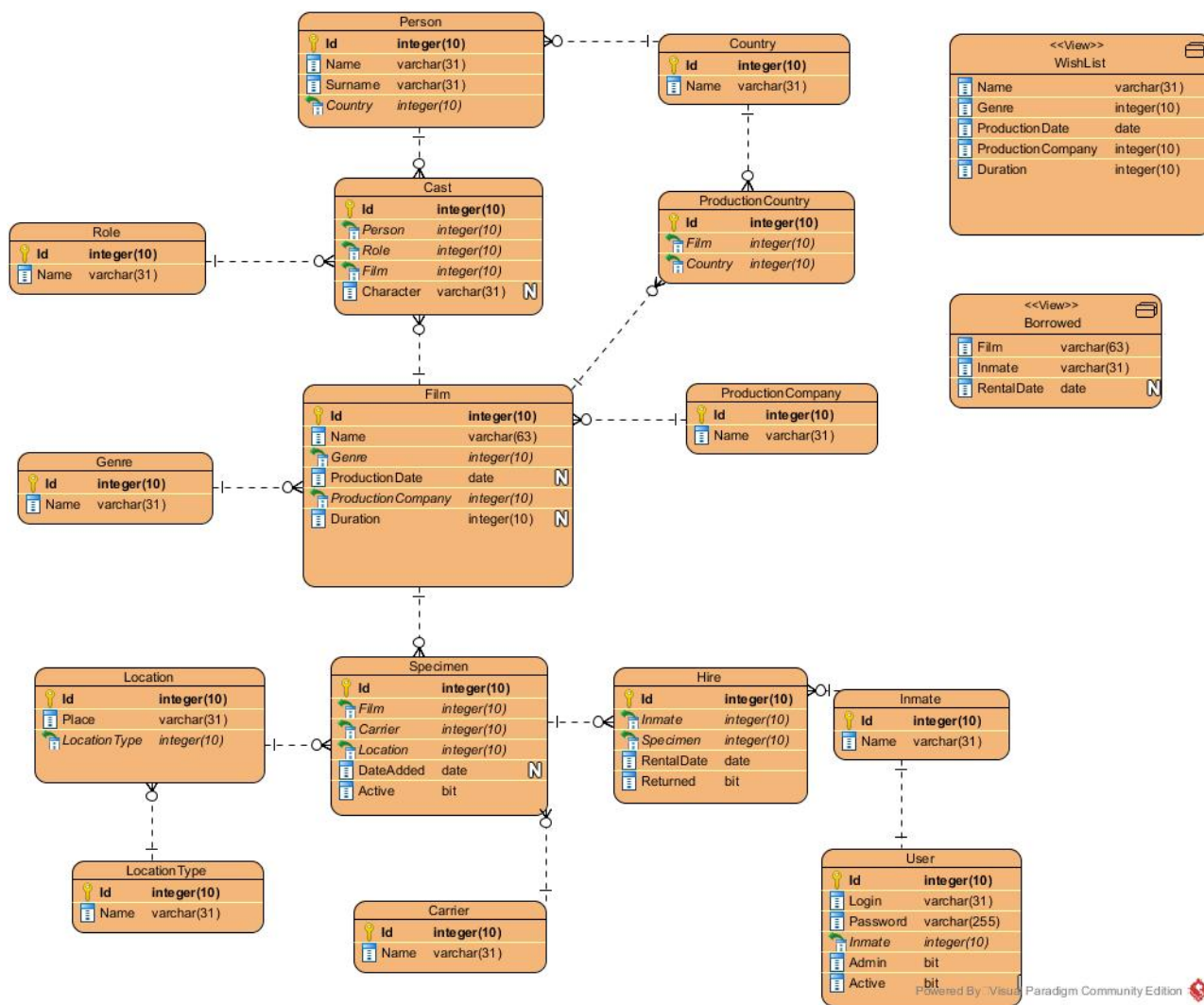
W diagramie przypadków użycia, podzieliliśmy funkcjonalności programu na klasy (Użytkownicy, Filmy, Tytuły, etc.) i wystosowaliśmy dla każdego obszaru po kilka funkcjonalności, po między którymi zachodzą relacje Include i Extend. Diagram w pełnej rozdzielczości można znaleźć w programie załączonym do niniejszego sprawozdania.



Rysunek 2: Diagram przypadków użycia

## 6 Diagram związków encji

Podczas tworzenia diagramu związków encji, kierowaliśmy się przedstawionym wcześniej modelem konceptualnym. Diagram związków encji, w stosunku do modelu konceptualnego został uzupełniony o typy danych dla poszczególnych pól w tabelach.



Rysunek 3: Diagram związków encji

## 7 Analiza ilości encji

### 7.1 Analiza liczby instancji dla każdej encji

- Film – ok. 50 rekordów (przykładowo)
- Specimen – ok. 50 rekordów (podobnie jak Film)
- Carrier (nośnik) – 5-10 rekordów (Nie więcej jak Specimen)
- LocationType (typ lokacji) – 5-10 rekordów (nie więcej jak Location)
- Location (lokacja) – ok. 50 rekordów (nie więcej jak Specimen)
- Hire (wypożyczenie) – 10 na sam początek, nie więcej jak 3 \* liczba użytkowników
- Inmate (domownik) – ok. 10 rekordów
- User (użytkownik) – ok. 10 rekordów (tak samo jak Inmate)
- ProductionCompany (Wytwórnia filmów) – ok. 10-15 rekordów
- Person (osoba) – minimalnie ilość rekordów 1 per Film (przykładowo)
- Cast (obsada) – maksymalnie Person \* Role \* Film
- Role (rola w filmie) – ok. 10 rekordów (przykładowo)
- Contry (kraj) – maksymalnie ok. 240, minimalnie 1
- ProductionCountry (Kraj produkcji) – maksymalnie Contry \* Film
- Genre (gatunek) – ok. 15 rekordów (przykładowo)

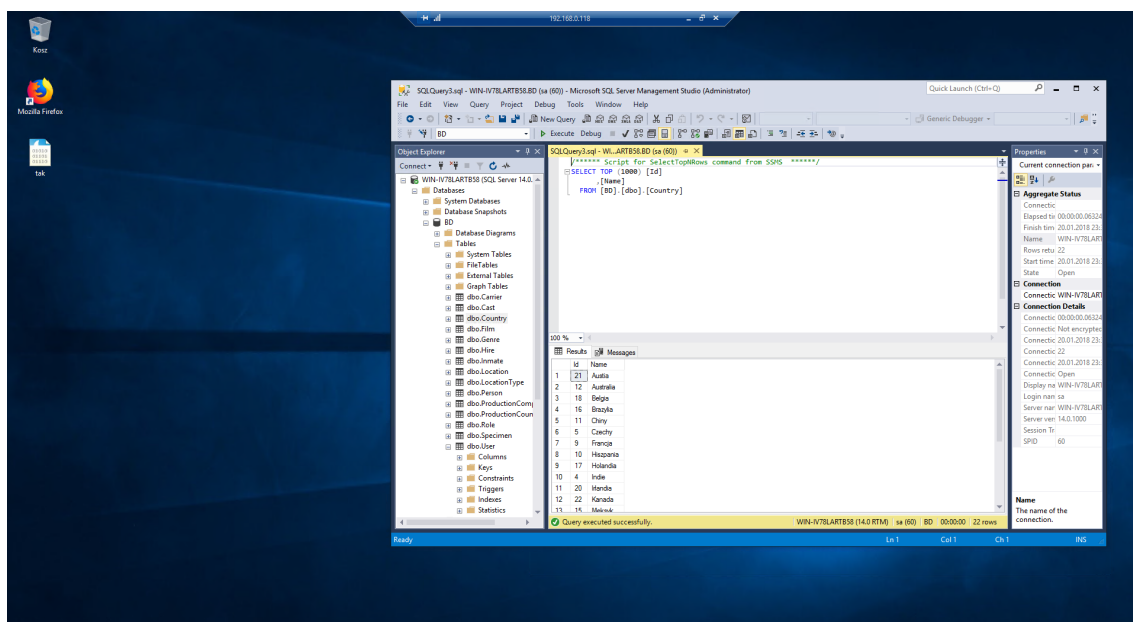
### 7.2 Analiza użycia dla każdej encji

- Film – wyszukiwanie najczęściej, potem dodawanie i edycja, prawie wcale usuwanie
- Specimen – wyszukiwanie najczęściej, potem dodawanie i edycja, prawie wcale usuwanie
- Carrier – wyszukiwanie najczęściej, rzadziej dodawanie (najwięcej rekordów na początku istnienia bazy), prawie wcale edycja, brak usuwania
- LocationType – wyszukiwanie najczęściej, potem dodawanie, brak usuwania i edycji
- Location – wyszukiwanie najczęściej, podobnie dodawanie, najmniej edycja, brak usuwania
- Hire – wyszukiwanie i dodawanie najczęściej, podobnie usuwanie, brak edycji
- Inmate – wyszukiwanie najczęściej, rzadziej dodawanie, brak edycji i usuwania
- User – wyszukiwanie najczęściej, rzadziej dodawanie i edycja, brak usuwania
- ProductionCompany – wyszukiwanie najczęściej, potem dodawanie, brak edycji i usuwania



- Person – wyszukiwanie najczęściej, potem dodawanie, brak edycji i usuwania
- Cast – najczęściej wyszukiwanie, potem dodawanie, brak edycji, rzadko usuwanie
- Role – najczęściej wyszukiwanie, rzadko dodawanie (najwięcej rekordów na początku istnienia bazy), brak edycji i usuwania
- Country – najczęściej wyszukiwanie, rzadko dodawanie, (najwięcej rekordów na początku istnienia bazy) brak edycji i usuwania
- ProductionCountry - najczęściej wyszukiwanie, potem dodawanie, rzadko edycja, brak usuwania
- Genre – najczęściej wyszukiwanie, rzadko dodawanie (najwięcej rekordów na początku istnienia bazy), brak usuwania i edycji

## 8 Implementacja bazy danych



Rysunek 4: Serwer na maszynie wirtualnej

### 8.1 Serwer bazodanowy

W związku z chęcią realizacji aplikacji na platformie .NET, zdecydowaliśmy się na wybór instancji bazodanowej dostarczonej przez firmę Microsoft, a mianowicie MS SQL Server 2017 w wersji Developer. Do jej zarządzania będziemy używać dostarczonego w pakiecie narzędzia - Management Studio, bądź też przeglądarki wbudowanej w środowisko programistyczne - Visual Studio 2015. Instancja serwera bazodanowego będzie umieszczona na maszynie wirtualnej (postawionej w środowisku VM VirtualBox) w systemie operacyjnym Microsoft Windows Server 2016 Standard. Aby umożliwić prostszą konfigurację,

dostęp do serwera będzie odbywał się z poziomu aplikacji Pulpitu Zdalnego. W celu użycia bazy danych w aplikacji na innym komputerze, odblokowano port 1433 na zaporze sieciowej serwera, odpowiadający za nasłuchiwanie połączeń przychodzących.

## 8.2 Tworzenie tabel i wprowadzanie danych

Tabele stworzono wg diagramów związków encji. Zawierają one przykładowe dane, niekoniecznie zgodne z rzeczywistością. Dla ułatwienia, tabeli zawierającej wypożyczenia, a także pola HashedPassword w tabeli Users nie wypełniono - będą one użyte, jak również wypełnione, podczas testów aplikacji w etapie III. Skrypt tabeli zamieszczony jest w załączniku.

## 9 Polityka bezpieczeństwa

### 9.1 Zabezpieczenie serwera

W celu zwiększenia bezpieczeństwa, dostęp do instancji bazodanowej będzie chroniony, tzn. aby się z nią połączyć potrzebne będzie hasło. Instancja posiada domyślny profil administratora (sa) oraz hasło dostarczone przez twórcę aplikacji (na potrzeby projektu, hasłem jest BazyDanych@2). Nie jest wymagane tworzenie dodatkowych kont do instancji.

### 9.2 Dostęp do kont użytkowników

Jak to zostało wspomniane w danych technicznych w punkcie 3.3, hasła użytkowników planujemy szyfrować. Myślimy o wykorzystaniu szyfrowania symetrycznego. Jako iż będziemy korzystać z platformy .NET, do naszych celów wykorzystamy gotową klasę CryptoStream biblioteki System.Security.Cryptography, która pozwoli nam przekształcać hasło, na niezrozumiały ciąg znaków. Wykorzystuje ona szyfrowanie *AES* (ang. Advanced Encryption Standard).

### 9.3 Sniffing

Przeprowadzono test bezpieczeństwa na serwerze z użyciem programu *Wireshark*. Celem było przechwycenie danych przychodzących lub wychodzących z instancji bazodanowej. Stwierdzono iż MSSQL Server przesyła dane za pomocą protokołu *TDS* (Tabular Data Stream) w formie odkrytej, tzn. za pomocą Wireshark'a jesteśmy w stanie przechwycić dane przychodzące (wysłane zdalnie zapytania), jak i wychodzące (wysyłane dane w zapytaniu). Test również wykazał iż nie jesteśmy w stanie określić hasła do serwera bazodanowego, tzn. część pakietów jest zaszyfrowana. Warto zaznaczyć również iż zabezpieczenie wskazane w poprzednim podpunkcie zabezpiecza hasło podczas przesyłania, tzn. jest ono przesyłane tylko w formie znanej aplikacji.

2818	127.077055	fe80::35c4:14c4:b05...	fe80::702f:65fb:c2:...	TDS	218 SQL batch
2819	127.077473	fe80::702f:65fb:c2:...	fe80::35c4:14c4:b05...	TDS	411 Response

> Frame 2818: 218 bytes on wire (1744 bits), 218 bytes captured (1744 bits) on interface 0					
> Ethernet II, Src: HonHaiPr_e3:5f:8f (ac:d1:b8:e3:5f:8f), Dst: PcsCompu_14:cf:eb (08:00:27:14:cf:eb)					
> Internet Protocol Version 6, Src: fe80::35c4:14c4:b05b:43f6, Dst: fe80::702f:65fb:c2:51a9					
> Transmission Control Protocol, Src Port: 53222, Dst Port: 1433, Seq: 2030, Ack: 2508, Len: 144					
> Tabular Data Stream					

0000	08 00 27 14 cf eb ac d1	b8 e3 5f 8f 86 dd 60 07	..'. .... .._...`.
0010	d9 17 00 a4 06 80 fe 80	00 00 00 00 00 00 35 c4	..... .....5.
0020	14 c4 b0 5b 43 f6 fe 80	00 00 00 00 00 00 70 2f	...[C... .....p/
0030	65 fb 00 c2 51 a9 cf e6	05 99 e1 44 9b 17 6b e3	e...Q... ...D.k.
0040	8c fd 50 18 08 13 8d 50	00 00 01 09 00 90 00 00	..P....P .....
0050	01 00 16 00 00 00 12 00	00 00 02 00 00 00 00 00	..... .....
0060	00 00 00 00 01 00 00 00	53 00 45 00 4c 00 45 00	..... S.E.L.E.
0070	43 00 54 00 20 00 5b 00	74 00 30 00 5d 00 2e 00	C.T. .[. t.0.]...
0080	5b 00 49 00 64 00 5d 00	2c 00 20 00 5b 00 74 00	[.I.d.]. ,. .[.t.
0090	30 00 5d 00 2e 00 5b 00	4e 00 61 00 6d 00 65 00	0.]...[. N.a.m.e.
00a0	5d 00 0d 00 0a 00 46 00	52 00 4f 00 4d 00 20 00	].....F. R.O.M. .
00b0	5b 00 64 00 62 00 6f 00	5d 00 2e 00 5b 00 47 00	[.d.b.o. ]...[.G.
00c0	65 00 6e 00 72 00 65 00	5d 00 20 00 41 00 53 00	e.n.r.e. ]. .A.S.
00d0	20 00 5b 00 74 00 30 00	5d 00	.[.t.0. ].

Rysunek 5: Odebrany pakiet (zapytanie)

2819	127.077473	fe80::702f:65fb:c2:...	fe80::35c4:14c4:b05...	TDS	411 Response
------	------------	------------------------	------------------------	-----	--------------

> Frame 2819: 411 bytes on wire (3288 bits), 411 bytes captured (3288 bits) on interface 0					
> Ethernet II, Src: PcsCompu_14:cf:eb (08:00:27:14:cf:eb), Dst: HonHaiPr_e3:5f:8f (ac:d1:b8:e3:5f:8f)					
> Internet Protocol Version 6, Src: fe80::702f:65fb:c2:51a9, Dst: fe80::35c4:14c4:b05b:43f6					
> Transmission Control Protocol, Src Port: 1433, Dst Port: 53222, Seq: 2508, Ack: 2174, Len: 337					
> Tabular Data Stream					

0000	ac d1 b8 e3 5f 8f 08 00	27 14 cf eb 86 dd 60 00	...._... '.....`.
0010	00 00 01 65 06 80 fe 80	00 00 00 00 00 00 70 2f	...e.... .....p/
0020	65 fb 00 c2 51 a9 fe 80	00 00 00 00 00 00 35 c4	e...Q... .....S.
0030	14 c4 b0 5b 43 f6 05 99	cf e6 6b e3 8c fd e1 44	...[C... ..k....D
0040	9b a7 50 18 08 13 af ae	00 00 04 01 01 51 00 3b	..P..... ..Q.;
0050	01 00 e3 03 00 12 00 00	81 02 00 00 00 00 00 00	..... .....
0060	00 10 00 38 02 49 00 64	00 00 00 00 00 08 00 a7	...8.I.d .....
0070	32 00 15 04 d0 00 00 04	4e 00 61 00 6d 00 65 00	2..... N.a.m.e.
0080	d1 01 00 00 00 09 00 41	6e 69 6d 6f 77 61 6e 79	.....A nimowany
0090	d1 02 00 00 00 0c 00 42	69 6f 67 72 61 66 69 63	.....B iografic
00a0	7a 6e 79 d1 03 00 00 00	07 00 4b 6f 6d 65 64 69	zny..... ..Komed
00b0	61 d1 04 00 00 00 06 00	48 6f 72 72 6f 72 d1 05	a..... Horror..
00c0	00 00 00 0a 00 50 72 7a	79 67 6f 64 6f 77 79 d1	.....Prz ygodowy.
00d0	06 00 00 00 0a 00 53 65	6e 73 61 63 79 6a 6e 79	.....Se nsacyjny
00e0	d1 07 00 00 00 0a 00 4b	72 79 6d 69 6e 61 6c 6e	.....K ryminaln
00f0	79 d1 08 00 00 00 0c 00	44 6f 6b 75 6d 65 6e 74	y..... Dokument
0100	61 6c 6e 79 d1 09 00 00	00 09 00 46 61 62 75 6c	alny.... ...Fabul
0110	61 72 6e 79 d1 0a 00 00	00 0a 00 4d 65 6c 6f 64	arny.... ...Melod
0120	72 61 6d 61 74 d1 0b 00	00 00 0e 00 50 6f 72 6e	ramat... ....Porn
0130	6f 67 72 61 66 69 63 7a	6e 79 d1 0c 00 00 00 07	ograficz ny.....
0140	00 57 6f 6a 65 6e 6e 79	d1 0d 00 00 00 0b 00 48	.Wojenny .....H
0150	69 73 74 6f 72 79 63 7a	6e 79 d1 0e 00 00 00 07	istorycz ny.....
0160	00 57 65 73 74 65 72 6e	d1 0f 00 00 00 06 00 44	.Western .....D
0170	72 61 6d 61 74 d1 10 00	00 00 05 00 41 6b 63 6a	ramat... ....Akcj
0180	61 d1 11 00 00 00 06 00	53 63 69 2d 46 69 fd 10	a..... Sci-Fi..

	Ethernet: <live capture in progress>
--	--------------------------------------

Rysunek 6: Wysłany pakiet (dane w odpowiedzi na zapytanie)

Podsumowując podpunkt, zabezpieczenia zastosowane w programie są wystarczające, tzn. nawet w

przypadku sniffingu nie jesteśmy w stanie odczytać wrażliwych danych (hasła) ze względu na zastosowane zabezpieczenia przy budowie projektu (haszowanie hasła) i przez protokoły serwera bazodanowego (szyfrowanie wrażliwych pakietów).

Należy też zauważyć, iż dostęp do bazy danych z poza sieci domowej (lokalnej) nie będzie możliwy (aplikacja pracuje tylko w sieci lokalnej), a co za tym idzie, ryzyko włamania do bazy danych bliskie zeru.

## 10 Aplikacja desktopowa

### 10.1 Środowisko i wykonanie

Program został napisany w środowisku Visual Studio 2015 na platformie .NET z użyciem języka C# i interfejsu Windows Forms. Do komunikacji z bazą danych używamy technologii LINQ to SQL, mapującej bazy danych relacyjne na obiektowe. Znaczy to mniej więcej, że mając bazę danych opartą na relacjach, jesteśmy w stanie odwoływać się do poszczególnych rekordów tabel jak do obiektów poszczególnych klas. Poprzez odwołanie do klasy *Database* jesteśmy w stanie wydobyć za pomocą chociażby wyrażenia lambda interesujący nas zestaw rekordów. Dla przykładu:

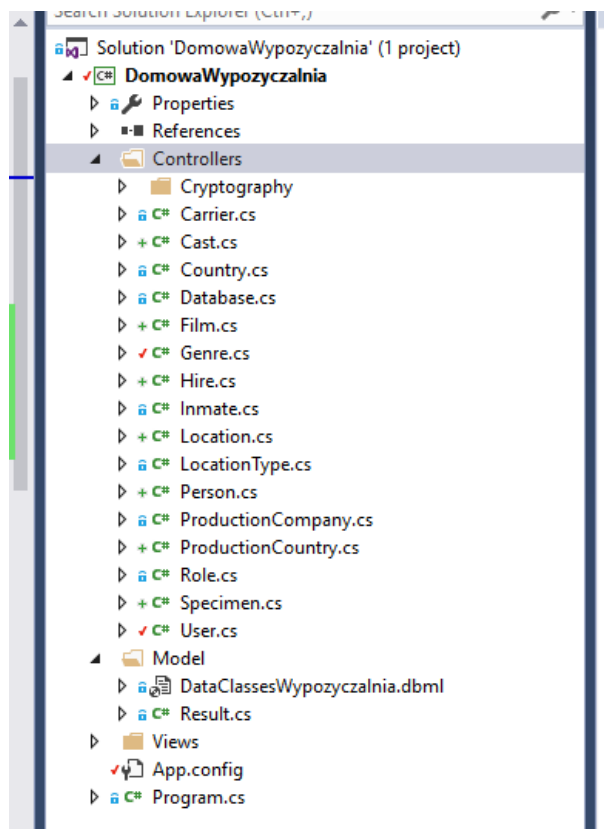
```
Database.Current.Genres.Where(g => g.Name.ToLower() == name.ToLower()).ToList();
```

wyrażenie lambda zwraca takie obiekty klasy *Genres*, których nazwa (*Name*) reprezentowana przez małe litery jest taka sama jak zadany parametr *name*.

### 10.2 Struktura projektu

Projekt jest podzielony wg wzorca *MVC* (Model-View-Controller). Są to 3 warstwy, które są odpowiedzialne za strukturę aplikacji. Każda z warstw posiada swój osobny folder, w którym są wydzielone odpowiednie klasy. W przypadku klas z Modelem i Kontrolera, wykorzystujemy klasy typu *partial*, jest są to pojedyncze klasy, ale rozbite na kilka plików.

- Model odpowiada za warstwę danych, ich struktury. Są to m.in. klasy wygenerowane przez plik *.dbml* oraz stworzony enum *Result*.
- Controller jest odpowiedzialny za przetwarzanie danych. Warstwa ta pobiera i przetwarza dane z bazy danych i wysyła odpowiednie odpowiedzi do warstwy Widoku.
- View (Widok) odpowiada za graficzną prezentację, a więc w tej warstwie znajdować się będą wszystkie definicje okien i akcje związane z posiadanymi kontrolkami.



Rysunek 7: Struktura projektu wg wzorca MVC

### 10.3 Połączenie z bazą danych

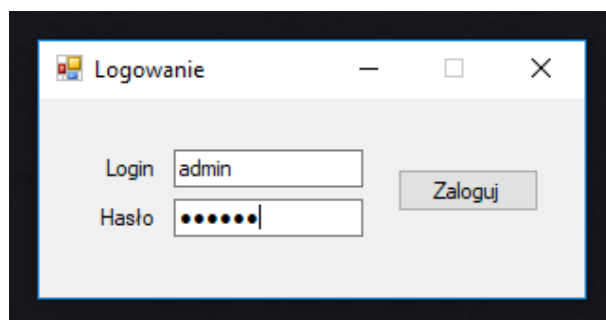
Aby się połączyć z bazą danych, w pliku *app.config* został umieszczony *ConnectionString*, który posiada podstawowe dane nt. łączenia się z bazą danych.

```
<connectionStrings>
  <add name="DomowaWypożyczalnia.Properties.Settings.BDConnectionString"
    connectionString="Data Source=WIN-IV78LARTB58;Initial Catalog=BD;
      User ID=sa;Password=BazyDanych@2;"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

Gdzie sekcja *name* to jest nazwa tego *ConnectionString*, sekcja *connectionString* posiada informacje, kolejno; nazwę instancji bazodanowej, nazwę bazy danych, a także login i hasło użytkownika. W sekcji *providerName* deklarujemy natomiast, że instancja bazodanowa jest dostarczona przez Microsoft i korzystamy z przestrzeni nazw *System.Data.SqlClient*.

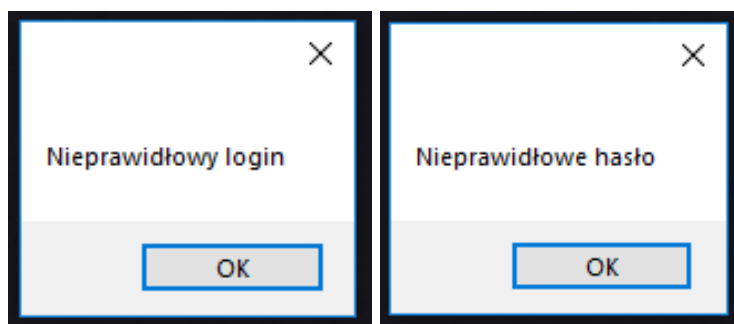
### 10.4 Logowanie do aplikacji

Logowanie do aplikacji odbywa się za pomocą loginu i hasła. Pole hasło ma ustawioną właściwość *PasswordChar* na *true*, a co za tym idzie - hasło jest niewidoczne podczas jego wpisywania.



Rysunek 8: Okno logowania

Tak zatwierdzone login i hasło są sprawdzane przez program pod kątem poprawności. Jeśli nie istnieje użytkownik o zadanym loginie, wyświetlany jest stosowny komunikat. W przypadku gdy użytkownik istnieje, ale hasło nie jest poprawne, wyświetlany jest komunikat o innej treści. Użytkownik oznaczony w bazie danych jako nieaktywny, a pragnący się zalogować do systemu, otrzyma komunikat o niepoprawnym loginie - próba zalogowania się takiej osoby, traktowana jest jak wpisanie nieprawidłowego loginu.

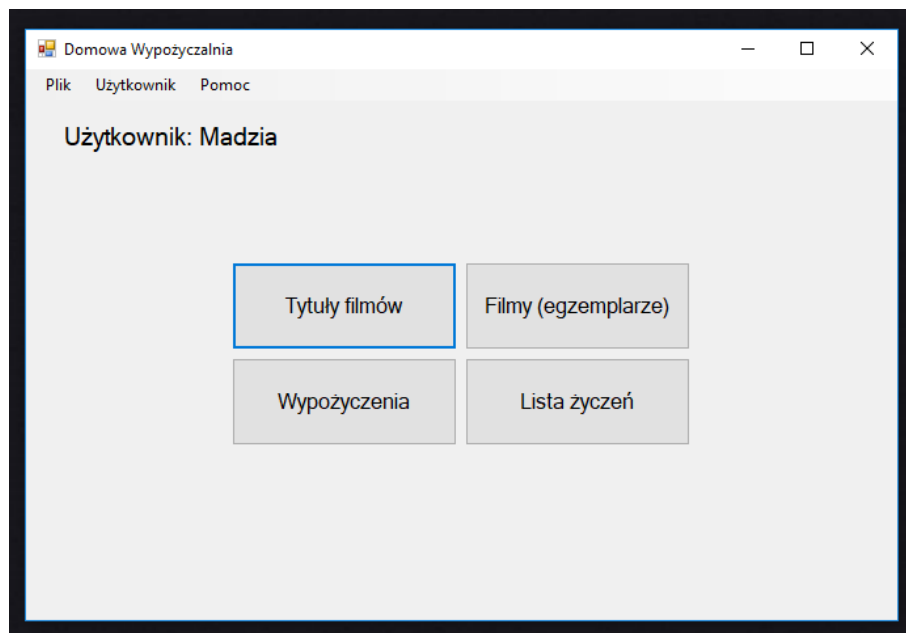


(a) Niepoprawnie wpisany login      (b) Niepoprawnie wpisane hasło

Rysunek 9: Komunikaty o braku poprawności

## 10.5 Okno główne

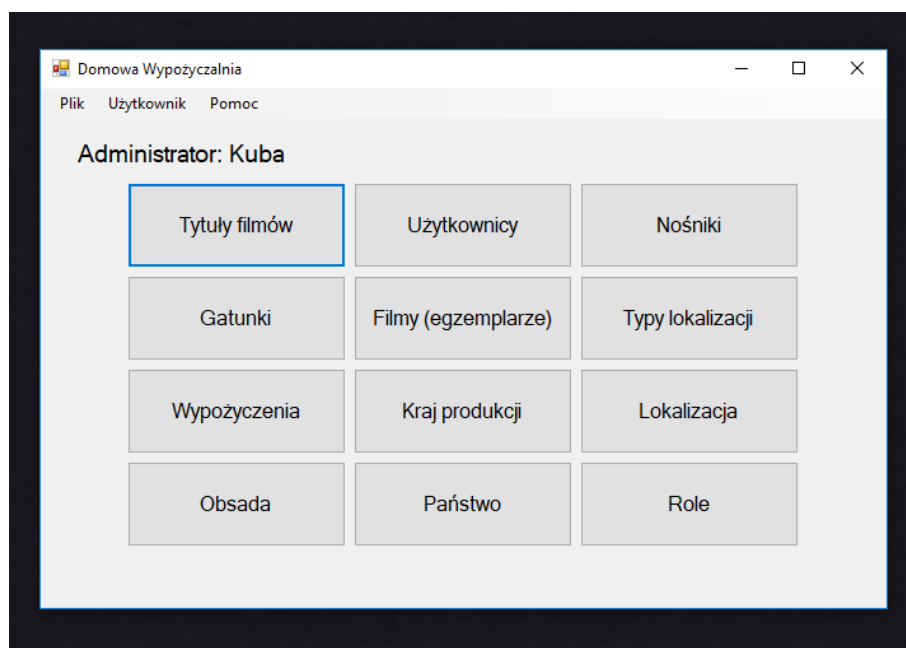
### 10.5.1 Panel użytkownika



Rysunek 10: Interfejs użytkownika

Jak widać, użytkownik ma wgląd w tytuły filmów, w poszczególne egzemplarze filmów, w aktualne (swoje) wypożyczenia i w listę życzeń, czyli te tytuły filmów, które nie posiadają egzemplarzy. W menu, na pasku, posiadamy pozycje: Plik (a tam opcję Zamknij), Użytkownik (zmień dane) oraz Pomoc (o programie).

### 10.5.2 Panel administratora



Rysunek 11: Interfejs administratora

Administrator posiada podobnie jak użytkownik, wgląd w Tytuły filmów, w same Filmy, ale też w Wypożyczenia wszystkich użytkowników oraz samych użytkowników. Może przeglądać, dodawać, usuwać i edytować Nośniki, Gatunki, Lokalizacje, Państwa, Kraje produkcji, Obsady itd. Ponadto na pasku menu posiada to samo co normalny użytkownik.

## 10.6 Funkcjonalności

Jeśli chodzi o poszczególne opcje dostępne w programie:

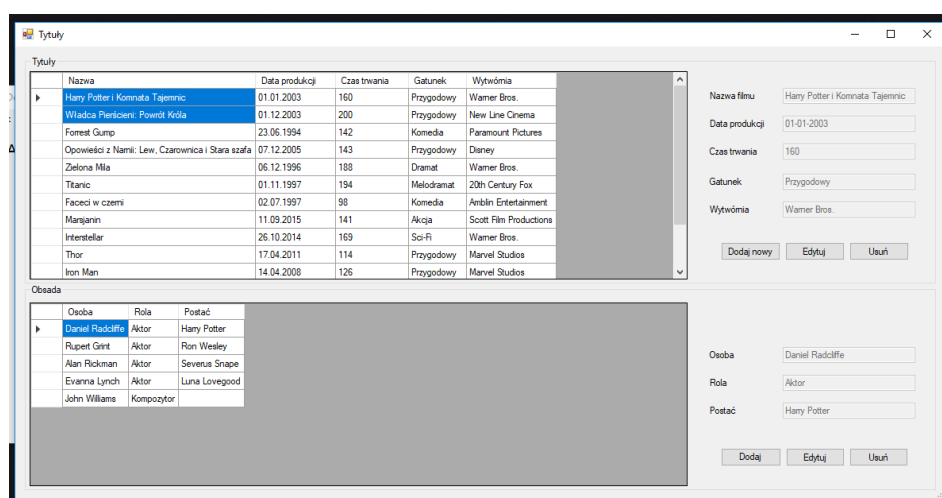
- Tytuły filmów - w oknie dostępna jest możliwość wyświetlenia, a także dodania nowych tytułów filmowych, a także aktorów. Tylko administrator posiada prawo edycji tytułu filmowego i aktora.
- Użytkownicy - opcja dostępna u administratora. Może resetować hasło dowolnych użytkowników, oznaczać ich jako nieaktywnych.
- Nośniki - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie dowolnego nośnika.
- Gatunki (filmowe) - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie dowolnego gatunku.
- Filmy (egzemplarze) - w oknie jest możliwość dodania nowych egzemplarzy istniejących już tytułów. Administrator może je edytować i oznaczać jako nieaktywne.
- Typy lokalizacji - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie dowolnego typu lokalizacji.



- Wypożyczenia - umożliwia przeglądanie aktywnych wypożyczeń i dokonanie zwrotu wypożyczonej pozycji. Administrator może przeglądać wypożyczenia wszystkich użytkowników i dokonywać administracyjnego zwrotu.
- Kraj produkcji - opcja dostępna u administratora. Umożliwia przypisanie kraju do dowolnego filmu jako kraju produkcji. Wyświetla wszystkie powiązania.
- Lokalizacja - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie lokalizacji.
- Obsada - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie aktorów przypisanych do danych filmów.
- Państwo - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie państwa z bazy danych.
- Role - opcja dostępna u administratora. Umożliwia wyświetlenie, dodanie, edycję i usunięcie możliwych ról w filmach.

## 10.7 Tytuły filmów

Okno posiada dwa komponenty typu *dataGridView*, umieszczonych w różnych *groupBox*ach. W pierwszym, znajdziemy wszystkie filmy z naszej bazy danych w kolejności chronologicznej jeśli chodzi o ich dodanie. W zależności od zaznaczonego wiersza, w polach obok będą się zmieniać dane. W drugim komponencie ukażą się nam aktorzy grający w danym filmie, oczywiście jeśli tacy występują. Podobnie tutaj po prawej stronie mamy pola, które będą nam pokazywać parametry danej postaci z filmu. Dodatkowo, mamy możliwość dodania nowego filmu, a także nowego aktora do filmu, aktualnie widnieje w zaznaczeniu. Administrator może wyedytować zaznaczoną postać i dany film oraz je usunąć. W przypadku usunięcia filmu szukamy, czy dany film nie posiada powiązań z innym rekordem. Jeśli nie ma ich, możemy śmiało usuwać rekord.



Rysunek 12: Stworzone okno z tytułami filmów

## 11 Testy aplikacji

Testy przeprowadziliśmy na bazie oraz w aplikacji. Polegały one na tym, że wpisywaliśmy wybrane dane. W bazie danych wyjątki były wyrzucane, kiedy były zostawione puste pola w miejscach, gdzie było zaznaczone "not null" oraz kiedy w miejsce integerów były wpisywane wyrazy. W aplikacji testy wyglądały podobnie, tylko że wszystkie te wyjątki zostały wylapane, a użytkownik został poinformowany, że należy zmienić format tekstu lub że nie może zostawić pustego pola.

## 12 Podsumowanie

Nasz projekt był czasochłonny. Należało najpierw przemyśleć, co chcemy zrobić, potem wykonać diagramy, zaprojektować bazę danych, napisać aplikację oraz sprawdzić, czy metody zostały poprawnie zaimplementowane. Mogliśmy się nauczyć, jak wygląda tworzenie aplikacji oraz co potrzebujemy do jej wykonania. Wartościowe było poznanie możliwości haszowania haseł oraz podejrzenia danych w Wiresharku, jaki to ma skutek. Do pozytywnych rzeczy możemy także zaliczyć stawianie bazy danych na maszynie wirtualnej. Mamy nadzieję, że trudności na jakie się natchnęliśmy oraz zdobyta wiedza pomogą nam w przyszłości na tworzenie lepszych projektów.

Nie zamieszczamy żadnej literatury i źródeł, ponieważ z nich nie korzystaliśmy. Nasz projekt powstał na bazie wiedzy nabytej podczas toku studiów oraz w czasie pracy.