

POLITECHNIKA WROCŁAWSKA
KATEDRA INFORMATYKI TECHNICZNEJ

BAZY DANYCH

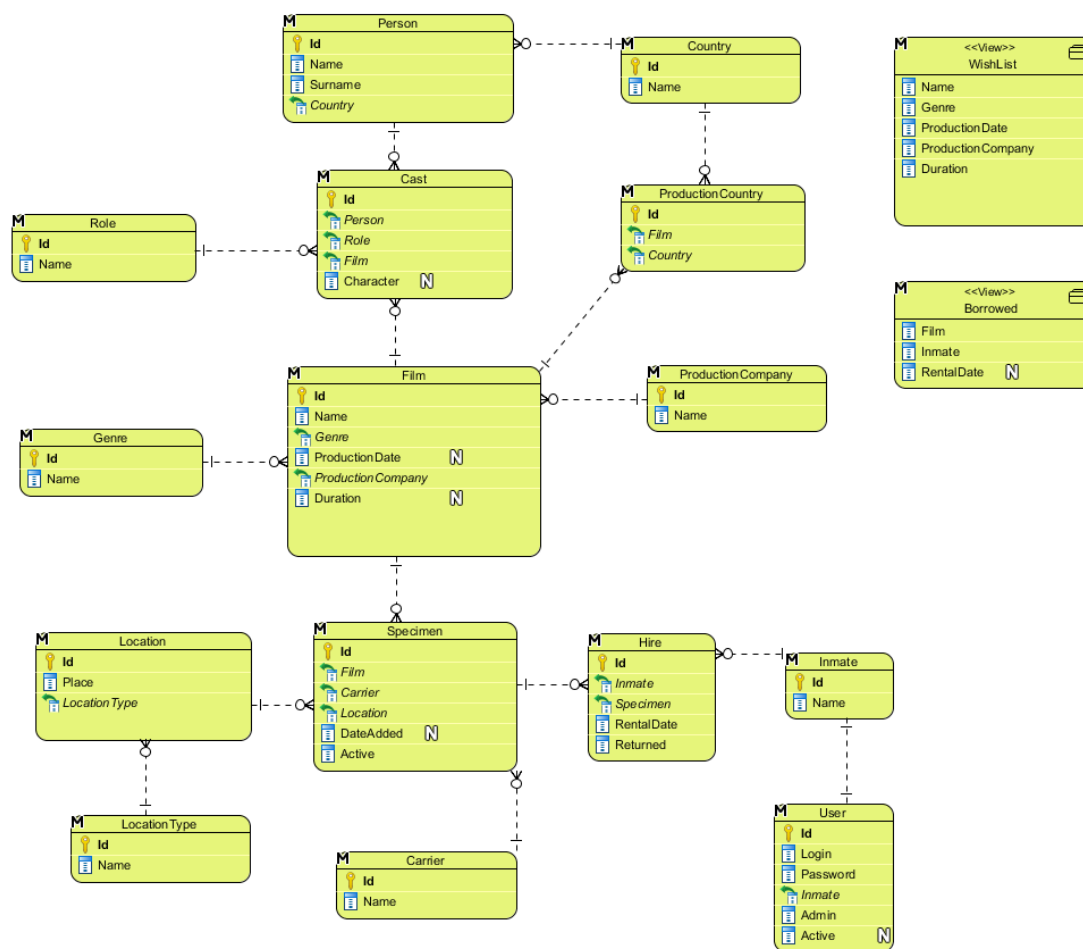
Magdalena Biernat

Mateusz Bortkiewicz

Prowadzący
dr inż. Tomasz Janiczek

21 grudnia 2017

1 Model konceptualny bazy danych



Rysunek 1: Model konceptualny

2 Opis "świata rzeczywistego"

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

2.1 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, mieszkanec 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.

2.2 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.

2.3 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 zahaszkowane hasło).

3 Wymagania funkcjonalne i нефункционале

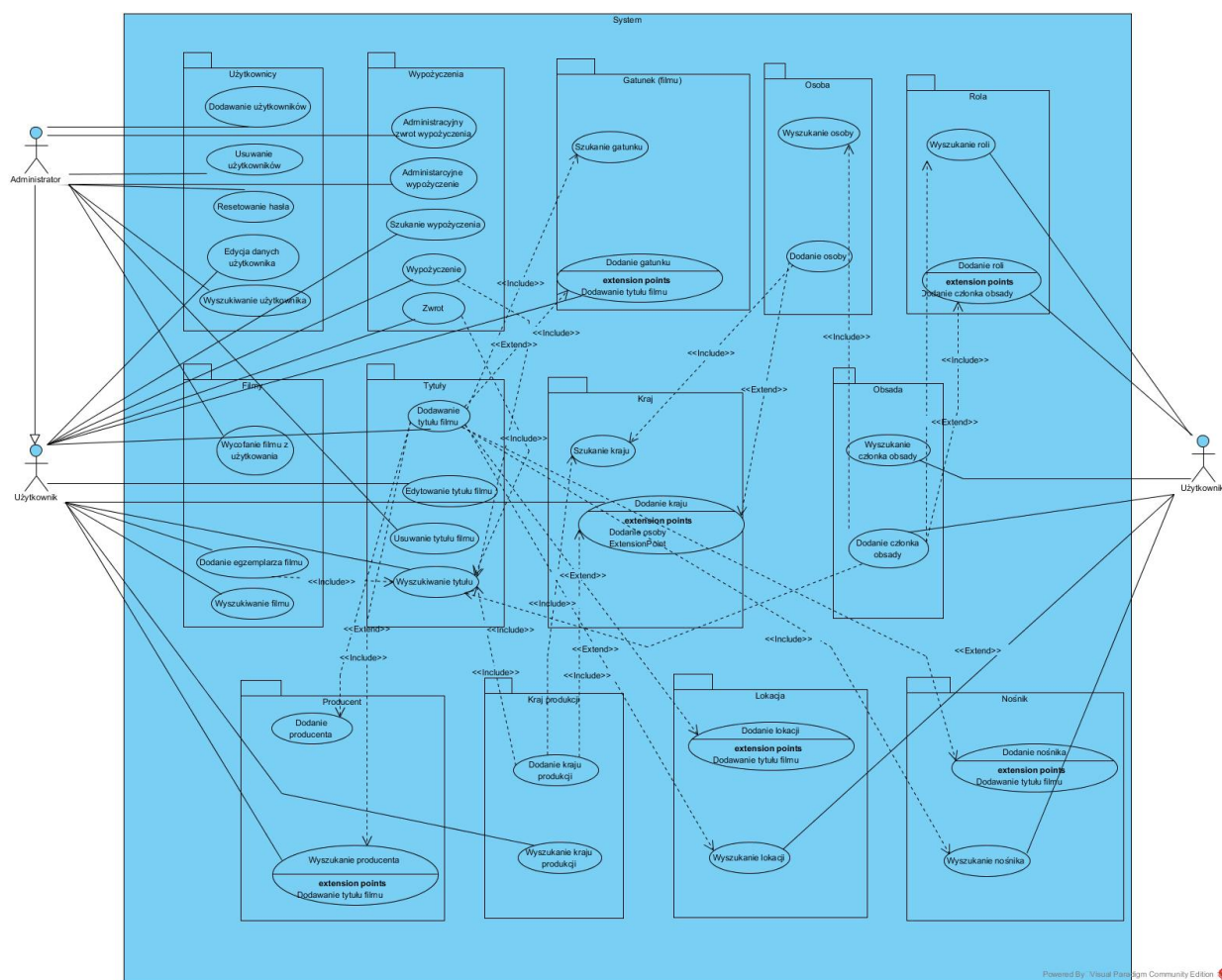
3.1 Wymagania funkcjonalne

- aplikacja ma mieć możliwość wyszukiwania\$dodawania\$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\$dodawania\$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)

3.2 Wymagania нефункционале

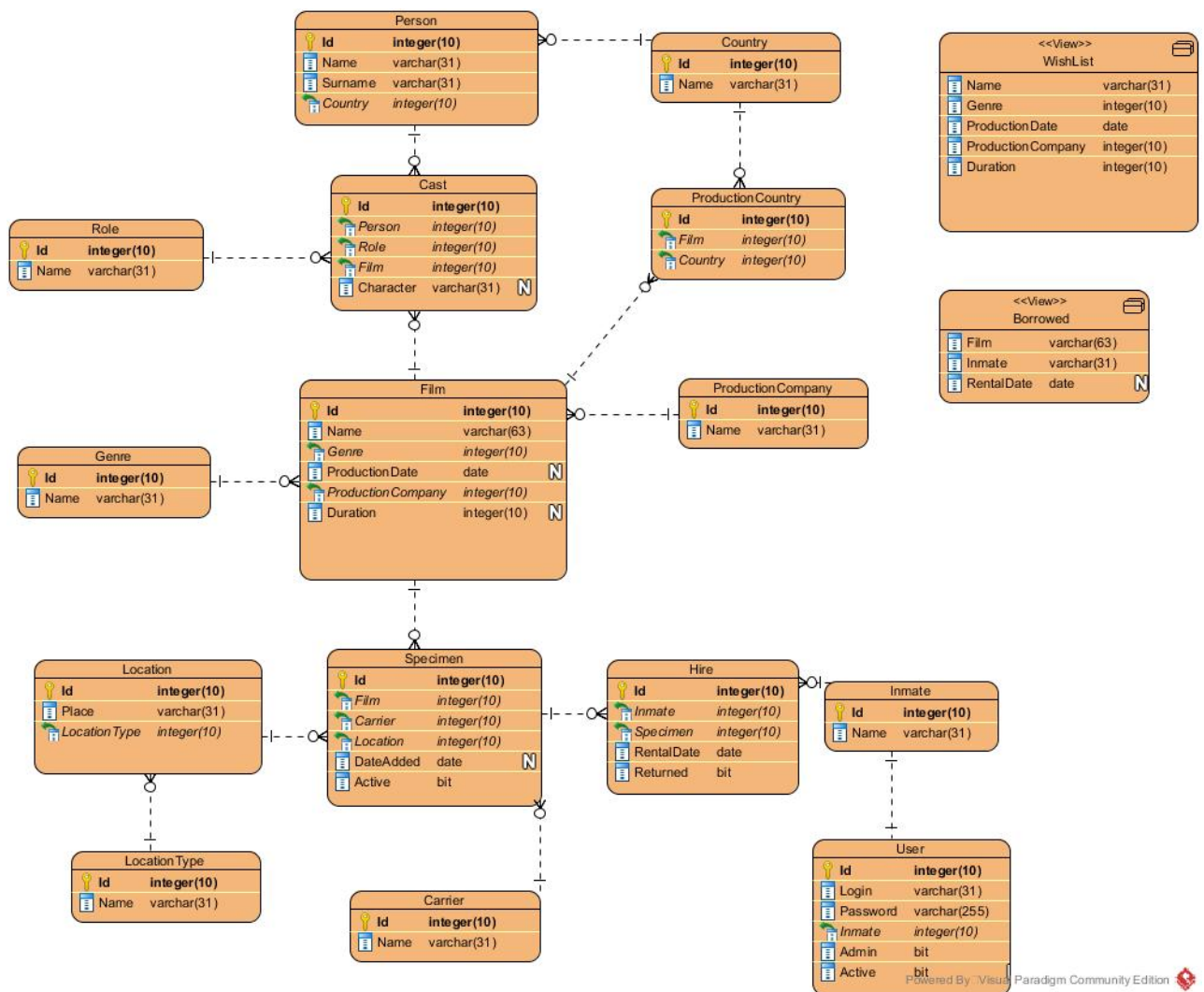
- ilość domowników nie wpływa na szybkość działania systemu
- ilość filmów/tytułów/użytkowników etc. Jest ograniczona tylko pojemnością dysku na którym stoi baza danych
- przyjazny interfejs użytkownika
- liczba błędów w aplikacji przez pierwszy miesiąc od wydania aplikacji nie może przekroczyć 5

4 Diagram przypadków użycia



Rysunek 2: Diagram PU

5 Diagram związków encji



Rysunek 3: Diagram związków encji

6 Analiza ilości encji

6.1 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)

6.2 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

7 Implementacja bazy danych

7.1 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 w wersji 2014. Do jej zarządzania będziemy używać dostarczonego w pakiecie narzędzia - Management Studio, bądź też przeglądarki wbudowanej w środowisko programistyczne - Visual Studio 2015.

7.2 Tworzenie tabel i insertowanie 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.

8 Polityka bezpieczeństwa

8.1 Zabezpieczenie serwera

W celu zwiększenia bezpieczeństwa, dostęp do instancji bazodanowej będzie chroniony, tzn. aby się z nią połączyć potrzebne będzie hasło. Na potrzeby budowania aplikacji i testów, instancja bazodanowa nie będzie posiadać hasła.

8.2 Dostęp do kont użytkowników

Jak to zostało wspomniane w danych technicznych w punkcie 2.3, 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).