

Bazy danych 2

Wypożyczalnia filmów

Termin zajęć: Poniedziałek 9:15-11:00

Prowadzący zajęcia: dr inż. Roman Ptak, W4/K9

Autor: Adam Filipowicz 221713

Spis treści

1	Wstęp	4
1.1	Temat	4
1.2	Cel projektu	4
1.3	Zakres projektu	4
2	Analiza wymagań	4
2.1	Opis działania	4
2.1.1	Lista wymagań funkcjonalnych	5
2.1.2	Lista wymagań niefunkcjonalnych	6
2.2	Harmonogram dalszej pracy nad projektem	6
3	Projekt systemu	7
3.1	Projekt bazy danych	7
3.1.1	Analiza rzeczywistości i uproszczony model konceptualny	7
3.1.2	Model logiczny	11
3.1.3	Model fizyczny	12
3.1.4	Mechanizmy przetwarzania danych	13
3.1.5	Projekt mechanizmów bezpieczeństwa na poziomie ba- zy danych	14
3.2	Projekt aplikacji użytkownika	15
3.2.1	Architektura aplikacji	15
3.2.2	Interfejs graficzny i struktura menu	16
3.2.3	Diagram przypadków użycia	20
3.2.4	Wybrane diagramy czynności	21
3.2.5	Metoda podłączania do bazy danych – integracja z ba- zą danych	22
3.2.6	Zabezpieczenia na poziomie aplikacji	23
4	Implementacja systemu baz danych	24
4.1	Tworzenie tabel i definiowanie ograniczeń	24
4.2	Implementacja mechanizmów przetwarzania danych	26
4.2.1	Widoki	26
4.2.2	Indeksy	26
4.3	Implementacja uprawnień i innych zabezpieczeń	26
4.3.1	Użytkownik	26
4.3.2	Pracownik	27

4.4	Testowanie bazy danych na przykładowych danych	28
4.4.1	Dodawanie prawidłowych danych i wygląd tabeli po dodaniu	28
4.4.2	Przykładowe dodawanie błędnych danych	36
4.4.3	Sprawdzenie poprawności działania widoków	37
4.4.4	Sprawdzenie uprawnień konta użytkownika korzysta- jącego z roli aplikacji użytkownika	38
5	Implementacja i testy aplikacji	39
5.1	Instalacja i konfigurowanie systemu	39
5.2	Instrukcja użytkowania aplikacji	39
5.3	Testowanie opracowanych funkcji systemu	41
5.4	Omówienie wybranych rozwiązań programistycznych	41
5.4.1	Implementacja interfejsu dostępu do bazy danych . . .	41
5.4.2	Implementacja wybranych funkcjonalności systemu . .	42
5.4.3	Implementacja mechanizmów bezpieczeństwa	42
6	Podsumowanie i wnioski	43

1 Wstęp

1.1 Temat

Wypożyczalnia filmów.

1.2 Cel projektu

Celem projektu jest zaprojektowanie i stworzenie bazy danych do wypożyczalni filmów umożliwiającej stosunkowo proste i intuicyjne zarządzanie filmami oraz użytkownikami stworzonej aplikacji. Ponadto baza danych ma umożliwić dodawanie oraz wgląd do wypożyczeń klientów. Kolejnym celem jest stworzenie aplikacji desktopowej na podstawie stworzonej bazy danych pozwalającej na korzystanie z niej sprzedawcom oraz klientom po stworzeniu konta.

1.3 Zakres projektu

Baza danych będzie umożliwiać gromadzenie, wyszukiwanie i przetwarzanie danych o filmach, klientach, sprzedawcach oraz wypożyczeniach. Aplikacja będzie wykorzystywać i aktualizować na bieżąco utworzoną bazę danych. Będzie pozwalała na modyfikację filmów (m.in. zmiana atrybutów przez sprzedawcę, usunięcie filmu) i kont (np. zmiana hasła).

2 Analiza wymagań

2.1 Opis działania

Aplikacja będzie pobierała dane z bazy m.in. przy logowaniu użytkowników, wyświetleniu listy filmów lub wyświetleniu wypożyczeń klientów. Modyfikacja danych w bazie będzie następować np. przy zmianie atrybutów konta albo dezaktywacji filmu.

Wypożyczalnia składa się z jednego ośrodka (siedziby firmy) znajdującego się we Wrocławiu. Serwer znajduje się w siedzibie firmy i jest jej własnością. Firma wypożycza filmy osobom prywatnym. Zatrudnieni będą pracować w siedzibie firmy. Film po wypożyczeniu przez klienta będzie dostępny przez okres jednego tygodnia.

Sprzedawca

Pracownik wypożyczalni (sprzedawca) może dodawać do bazy danych nowe filmy. Może je edytować i dezaktywować. Może także wyświetlać listę filmów. Każdy film jest reprezentowany przez następujące dane: nazwa, cena, rok powstania, typ filmu oraz opis. Sprzedawca może sprawdzić historię wypożyczeń danego klienta, danego filmu oraz całą, najnowszą historię wypożyczeń. Może także wyświetlić wszystkie konta utworzone w aplikacji. Może zmienić dane swojego konta: hasło, imię, nazwisko oraz adres.

Klient

Klient może założyć konto podając: nazwę konta, hasło, imię, nazwisko oraz adres. Może on wyświetlić listę filmów i wybrać film do wypożyczenia. Może przejrzeć historię swoich wypożyczeń, zmienić dane konta: hasło, imię, nazwisko oraz adres, a także dezaktywować swoje konto.

2.1.1 Lista wymagań funkcjonalnych

- Pracownik może dodać film do bazy podając unikalną nazwę oraz inne atrybuty.
- Pracownik może usunąć film z bazy przez podanie jego numeru ID.
- Pracownik może edytować atrybuty filmu przez podanie numeru ID oraz tych atrybutów.
- Pracownik może wyświetlić historię wypożyczeń dla wybranego klienta, wybranego filmu oraz najnowszą, pełną historię.
- Pracownik może wyświetlić wszystkie istniejące konta.
- Klient może wypożyczyć filmy. Płatność jest osobna dla każdego z filmów. Film, po zatwierdzonej płatności jest aktywny przez jeden tydzień.
- Klient może usunąć (dezaktywować) swoje konto.
- Klienci i pracownicy mogą przeglądać listę filmów.
- Klienci i pracownicy mogą tworzyć konta z unikalnymi nazwami.
- Klienci i pracownicy mogą edytować dane swojego konta.

2.1.2 Lista wymagań niefunkcjonalnych

- Wypożyczenia mogą być dokonane przez aplikację desktopową.
- Używana technologia: Java.
- Używanym i zalecanym systemem operacyjnym jest Windows.
- Czas ładowania aplikacji nie przekracza sekundy, a ładowanie wyników powinno być szybkie.
- Używana technologia baz danych: Microsoft SQL Server.

Wymagania dotyczące rozmiaru baz danych

Zakłada się, że klientów w bazie danych będzie kilkadziesiąt tysięcy, zatem liczba klientów jednocześnie przeglądających dane katalogu oraz dokonujących wypożyczeń będzie wynosiła około 1000. Wypożyczalnia może oferować kilkadziesiąt tysięcy tytułów. Liczba pracowników będzie zależała od sytuacji finansowej firmy, ale nie będzie przekraczała 10 osób. Zakłada się, że klient wypożyczy średnio 2 filmy miesięcznie, zatem ilość wypożyczeń miesięcznie będzie wynosiła około stu tysięcy. Dane dotyczące wypożyczeń będą składowane zgodnie z obowiązującym prawem przez okres pięciu lat.

2.2 Harmonogram dalszej pracy nad projektem

- zaprojektowanie i implementacja opisanej bazy danych
- implementacja i testowanie aplikacji połączonej z bazą danych i spełniającej listy wymagań funkcjonalnych oraz niefunkcjonalnych
- analiza oraz poprawa błędów aplikacji i bazy danych aż do skutku

3 Projekt systemu

3.1 Projekt bazy danych

3.1.1 Analiza rzeczywistości i uproszczony model konceptualny

W wypożyczalniach filmów online stosowane są bazy danych przechowujące dane głównie o posiadanych filmach, pracownikach, użytkownikach wypożyczalni oraz wypożyczeniach. Z danych tych można następnie analizować zachowania użytkowników aby poprawić jakość działania wypożyczalni oraz zwiększyć jej zysk (np. poprzez wycofywanie filmów które są wypożyczane rzadko lub nigdy, czy analiza powodów dezaktywacji konta przez klienta). Bezpośrednia ingerencja człowieka w bazę następuje przy pomocy stworzonej aplikacji przez:

- tworzenie, edycję i dezaktywację konta,
- prawidłowe wypożyczenie filmu przez klienta,
- dodawanie, edycję i usunięcie filmu przez pracownika.

Encje

- FILM - konkretny film do wypożyczenia wraz z opisującymi go atrybutami.
 - idF - unikalny numer filmu
 - nazwaF - nazwa filmu, może się powtarzać
 - cenaF - cena za tygodniowe wypożyczenie filmu
 - rokPowstaniaF - rok powstania filmu do łatwiejszego znalezienia w liście filmów
 - gatunekF - gatunek filmu, np. komedia, dramat
 - opisF - krótki opis do identyfikacji filmu
 - reżyserzyF - reżyserzy filmu
 - aktorzyF - aktorzy występujący w filmie

Nazwa atrybutu	Typ
idF	Liczba naturalna
nazwaF	Max. 50 znaków
cenaF	Liczba dodatnia
rokPowstaniaF	Liczba naturalna
gatunekF	Max. 60 znaków
opisF	Max. 1000 znaków
reżyserzyF	Max. 100 znaków
aktorzyF	Max. 300 znaków

- UŻYTKOWNIK - konto klienta lub pracownika do korzystania z aplikacji wypożyczalni.
 - idU - unikalny numer konta
 - nazwaU - unikalna nazwa konta
 - hasłoU - szyfrowane hasło do konta
 - imieU - imie użytkownika konta
 - nazwiskoU - nazwisko użytkownika konta
 - adresU - adres korespondencyjny użytkownika konta

Nazwa atrybutu	Typ
idU	Liczba naturalna
nazwaU	Max. 50 znaków
hasłoU	Liczba dodatnia
imieU	Max. 30 znaków
nazwiskoU	Max. 60 znaków
adresU	Max. 200 znaków

- PRACOWNIK
 - idP - id pracownika
 - zarobkiP - zarobki pracownika
 - dataZatrudnieniaP - data zatrudnienia pracownika

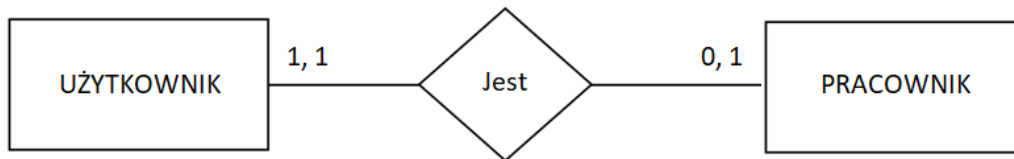
Nazwa atrybutu	Typ
idP	Liczba naturalna
zarobkiP	Liczba dodatnia
dataZatrudnieniaP	Data w formacie yyyyMMdd

- WYPOŻYCZENIE - pojedyncze wypożyczenie filmu przez klienta z informacjami potrzebnymi do analizy danych.
 - idW - unikalny numer wypożyczenia
 - cenaW - cena w momencie wypożyczenia
 - dataWypozyczeniaW - data w której film został wypożyczony
 - dataZakonczeniaW - data zakończenia wypożyczenia filmu (tydzień później od daty wypożyczenia)

Nazwa atrybutu	Typ
idW	Liczba naturalna
cenaW	Liczba dodatnia
dataWypozyczeniaW	Data w formacie yyyyMMdd
dataZakonczeniaW	Data w formacie yyyyMMdd

Związki

- Jest(UŻYTKOWNIK(1,1):PRACOWNIK(0,1))



- Pracownik musi być użytkownikiem.
- Dany pracownik jest dokładnie jednym użytkownikiem.
- Użytkownik nie musi być pracownikiem.
- Użytkownik może być jednym pracownikiem.

- ZawieraSie(UŻYTKOWNIK(1,1):WYPOŻYCZENIE(0,N))



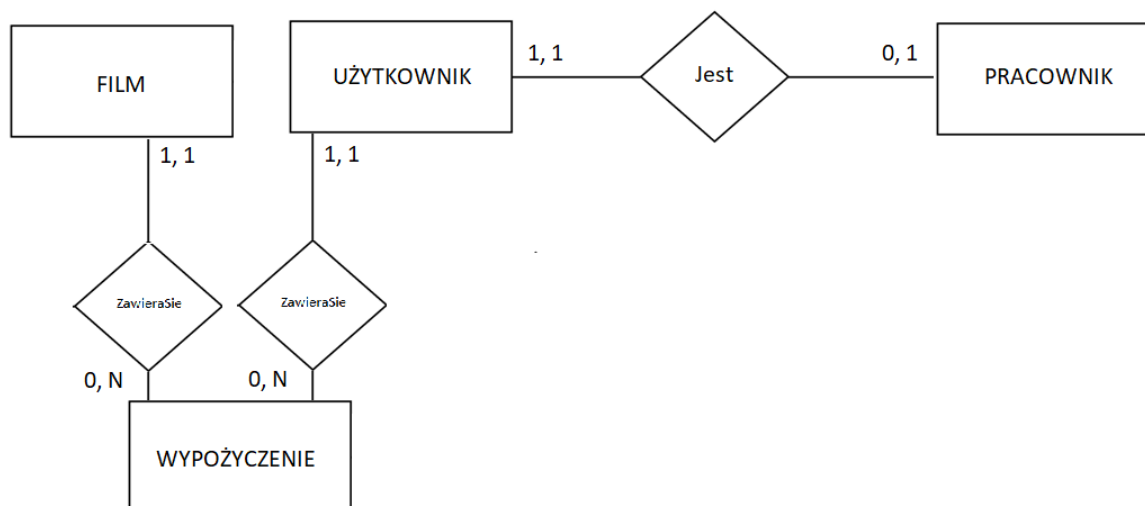
- Użytkownik nie musi zawierać się w żadnym wypożyczeniu.
- Dany użytkownik zawiera się w dowolnej liczbie wypożyczeń.
- Wypożyczenie musi zawierać użytkownika.
- Wypożyczenie zawiera dokładnie jednego użytkownika

- ZawieraSie(FILM(1,1):WYPOŻYCZENIE(0,N))



- Film nie musi zawierać się w żadnym wypożyczeniu.
- Dany film zawiera się w dowolnej liczbie wypożyczeń.
- Wypożyczenie musi zawierać film.
- Wypożyczenie zawiera dokładnie jeden film.

Diagram obiektowo-związkowy (ERD)



3.1.2 Model logiczny

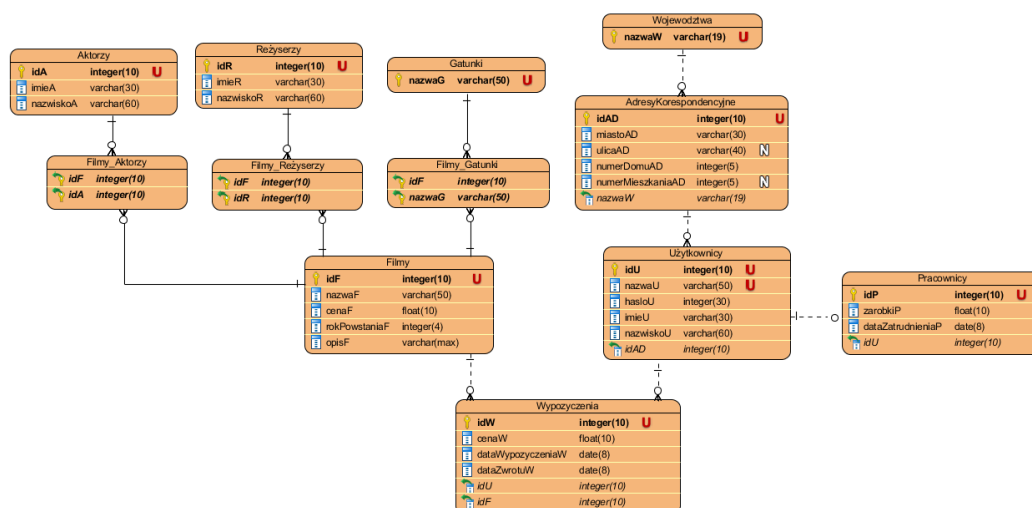
- Jest(UŻYTKOWNIK(1,1):PRACOWNIK(0,1))**
 UŻYTKOWNIK(idU, nazwaU, hasłoU, imieU, nazwiskoU, adresU)
 PRACOWNIK(idP, zarobkiP, dataZatrudnieniaP)
 Po przekształceniu otrzymujemy:
 Użytkownicy(idU, nazwaU, hasłoU, imieU, nazwiskoU, adresK)
 Pracownicy(idP, #idU, zarobkiP, dataZatrudnieniaP)
- ZawieraSie(UŻYTKOWNIK(1,1):WYPOŻYCZENIE(0,N))**
 UŻYTKOWNIK(idU, nazwaU, hasłoU, imieU, nazwiskoU, adresU)
 WYPOŻYCZENIE(idW, cenaW, dataWypożyczeniaW, dataZakończeniaW)
 Po przekształceniu otrzymujemy:
 Użytkownicy(idU, nazwaU, hasłoU, imieU, nazwiskoU, adresU)
 Wypożyczenia(idW, cenaW, dataWypożyczeniaW, dataZakończeniaW, #idK)

- ZawieraSie(FILM(1,1):WYPOŻYCZENIE(0,N)
FILM(idF, nazwaF, cenaF, rokPowstaniaF, gatunekF, opisF, reżyserzyF, aktorzyF)
WYPOŻYCZENIE(idW, cenaW, dataWypożyczeniaW, dataZakonczeniaW)
Po przekształceniu otrzymujemy:
Filmy(idF, nazwaF, cenaF, rokPowstaniaF, gatunekF, opisF, reżyserzyF, aktorzyF)
Wypożyczenia(idW, cenaW, dataWypożyczeniaW, dataZakonczeniaW, #idF)

3.1.3 Model fizyczny

Model fizyczny został wykonany przy użyciu programu Visual Paradigm 14.2. W wyniku normalizacji zostały dodane tabele: Reżyserzy, Aktorzy oraz Gatunki połączone odpowiednimi tabelami z tabelą Filmy. Adres w tabeli użytkownik został także rozbitý na tabelę słownikową "Województwa" oraz tabelę "AdresyKorespondencyjne" zawierającą: nazwę miasta, ulicy, numer domu oraz numer mieszkania.

Pełna normalizacja wymagałaby także rozbicia tabeli AdresyKorespondencyjne na tabelę słownikową Miasta połączoną z województwami 1 do wielu, oraz pozostawienia reszty pól w tabeli AdresyKorespondencyjne.



3.1.4 Mechanizmy przetwarzania danych

W projekcie użyję transakcyjnego przetwarzania danych. Jest ono konieczne między innymi przy potwierdzaniu płatności, żeby uniknąć przypadków gdy zrobimy przelew, a wypożyczony film nie będzie dostępny. Transakcje pomogą zachować integralność danych.

ID we wszystkich tabelach są kluczami podstawowymi oraz wartościami unikatowymi. Nazwa użytkownika także jest wartością unikatową, by zapewnić bezproblemowe logowanie do aplikacji. Nazwa gatunku filmu jest wartością unikatową, by zapobiec przypadkowemu umieszczeniu dwóch tych samych nazw gatunków (tabela gatunki jest tabelą słownikową). Numer mieszkania użytkownika jest jedyną wartością która może przyjmować wartości NULL, ponieważ jest przeznaczona tylko dla mieszkańców bloków, dla sprecyzowania mieszkania w bloku. Nazwa ulicy może być pusta w przypadku gdy w małej miejscowości ulica nie ma jeszcze nazwy.

Widoki z punktu widzenia użytkownika traktowane są tak samo jak zwykłe tabele. Mogą jednak łączyć różne atrybuty wielu tabel, dlatego są przydatne w przypadku wielokrotnego odwoływania się do potrzebnych danych. W moim projekcie zdefiniowałem następujące widoki:

Nazwa widoku	Atrybut	Tabela
WidokFilm	nazwaF	Film
	cenaF	Film
	rokPowstaniaF	Film
	opisF	Film
	imięR	Reżyserzy
	nazwiskoR	Reżyserzy
	imięA	Aktorzy
	nazwiskoA	Aktorzy
	nazwaG	Gatunki
WidokAdres	nazwaW	Wojewodztwa
	miastoAD	AdresyKorespondencyjne
	ulicaAD	AdresyKorespondencyjne
	numerDomuAD	AdresyKorespondencyjne
	numerMieszkaniaAD	AdresyKorespondencyjne

W celu usprawnienia wyszukiwania danych w tabelach wprowadzę do projektu indeksy:

- nazwaU z tabeli Uzytkownicy - atrybut używany zarówno przy logowaniu jak i wyszukiwaniu użytkownika oraz potrzebny do wypożyczeń
- nazwaF z tabeli Filmy - atrybut najczęściej używany do wyszukania potrzebnego filmu, a także do wypożyczenia
- rokPowstaniaF z tabeli Filmy - atrybut w drugiej kolejności najczęściej używany do wyszukania filmu
- dataWypozyczeniaW z tabeli Wypozyczenia - atrybut często używany przez pracownika do wyszukania wypożyczenia z danego dnia

3.1.5 Projekt mechanizmów bezpieczeństwa na poziomie bazy danych

Używanym środowiskiem będzie MS SQL. Domyślnie tworzy się w nim konto administratora systemu (sa) oraz właściciela bazy danych (dbo). Administrator przyznaje wszystkie przywileje i uprawnienia na bazy danych. Stworzone zostaną dwie role: User (dla użytkowników) oraz Worker (dla pracowników). Tabelka z przydzieleniem opcji grant poszczególnym rolam:

	User			Worker		
	Update	Select	Insert	Update	Select	Insert
Filmy		✓		✓	✓	✓
Uzytkownicy	✓	✓	✓	✓	✓	✓
Pracownicy					✓	
Wypozyczenia		✓	✓		✓	
Gatunki		✓		✓	✓	✓
Rezyserzy		✓		✓	✓	✓
Aktorzy		✓		✓	✓	✓
AdresyKorespondencyjne	✓	✓	✓	✓	✓	✓
Wojewodztwa		✓			✓	
Filmy_Gatunki		✓		✓	✓	✓
Filmy_Aktorzy		✓		✓	✓	✓
Filmy_Rezyserzy		✓		✓	✓	✓

3.2 Projekt aplikacji użytkownika

3.2.1 Architektura aplikacji

Na początku aplikacji jest wybór na menu klienta oraz sprzedawcy, żeby nie było konieczności pisania osobnej aplikacji dla sprzedawców. Bezpieczeństwo sprzedawcy nie jest naruszone, ponieważ do zalogowania potrzebne jest hasło.

Menu klienta

Po wybraniu menu klienta możliwe jest

- utworzenie konta po podaniu jego parametrów: nazwy konta, hasła, imienia, nazwiska oraz adresu.
- zalogowanie do konta po podaniu nazwy konta oraz hasła.

Po zalogowaniu wyświetlane są dane konta oraz domyślna lista filmów. Możliwa jest edycja danych konta oraz wyszukiwanie filmów po wpisaniu nazwy, typu lub roku powstania. Wyświetlanych jest 5 filmów na jednej stronie i jest możliwość przełączania stron. Możliwe jest wciśnięcie jednego z filmów i przekierowanie na stronę potwierdzenia wypożyczenia a także wylogowanie z konta.

Po przejrzaniu danych filmu przez klienta oraz potwierdzeniu wypożyczenia następuje przekierowanie do przelewu i możliwość oglądania filmu (jeśli przelew był poprawny).

Menu sprzedawcy

Po wybraniu menu sprzedawcy możliwe jest zalogowanie do konta po podaniu nazwy konta oraz hasła.

Po zalogowaniu wyświetlane są dane konta. Możliwa jest edycja danych konta, przejście do przeglądania wypożyczeń klientów, edycja filmów oraz wyświetlenie kont klientów a także wylogowanie.

Po wybraniu edycji filmów możliwe jest:

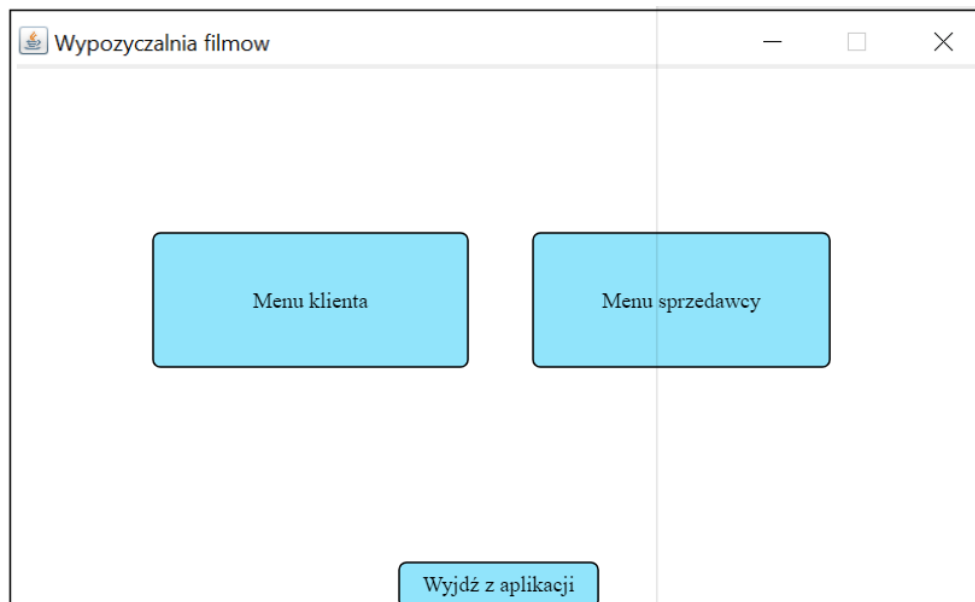
- znalezienie filmu przez wpisanie wybranych parametrów (najłatwiej przez wpisanie ID filmu)
- utworzenie filmu po wpisaniu wszystkich parametrów

- edycja filmu po wcześniejszym znalezieniu filmu i edycji niektórych parametrów
- usunięcie filmu po wcześniejszym jego znalezieniu
- przejście do następnego lub poprzedniego filmu jeśli w bazie danych znaleziono kilka filmów dla podanych wartości (np. nazwie)
- przeglądanie stron listy znalezionych filmów
- powrót do konta.

Po wybraniu przeglądania wypożyczeń wyświetlane są dane aktualnego wypożyczenia. Możliwe jest wyszukanie wypożyczenia po podaniu wybranych danych (najprościej po podaniu ID wypożyczenia) oraz przeglądanie wypożyczeń, gdy znaleziono kilka. Możliwy jest także powrót do menu pracownika.

3.2.2 Interfejs graficzny i struktura menu

Wszystkie mockupy zostały utworzone przez aplikację webową moqups.



Rysunek 1: Okno startowe - wybór menu klienta lub sprzedawcy

Rysunek 2: Menu klienta. Menu sprzedawcy różni się podpisami oraz brakiem możliwości utworzenia konta.

Rysunek 3: Konto klienta wraz z funkcjonalnością.



Rysunek 4: Okno po wybraniu filmu do wypożyczenia, wraz z opisem filmu.



Rysunek 5: Konto sprzedawcy wraz z funkcjonalnością.

Wypożyczalnia filmów

Edytuj filmy

Edycja filmów:

ID filmu:

Nazwa filmu:

Typ filmu:

Rok powstania:

Opis filmu:

Cena:

Gatunek:

Reżyser:

Aktor:

Lista filmów:

Przykładowy film 1

Przykładowy film 2

Przykładowy film 3

Przykładowy film 4

Przykładowy film 5

Strona 1

Poprzednia strona

Następna strona

Znajdź

Stwórz

Wróć do konta

Edytuj

Usuń

Rysunek 6: Okno do edycji filmów w bazie.

Wypożyczalnia filmów

Przejrzyj wypożyczenia

ID wypożyczenia: 205826

Nazwa filmu: Przykładowy film 2

Nazwa klienta: Konto przykładowe

Cena filmu: 20.0

Data wypożyczenia: 20.03.2017

Data zakończenia: 27.03.2017

Następne wypożyczenie

Poprzednie wypożyczenie

Wyszukiwarka wypożyczeń:

ID wypożyczenia:

Nazwa klienta:

Nazwa filmu:

Data wypożyczenia:

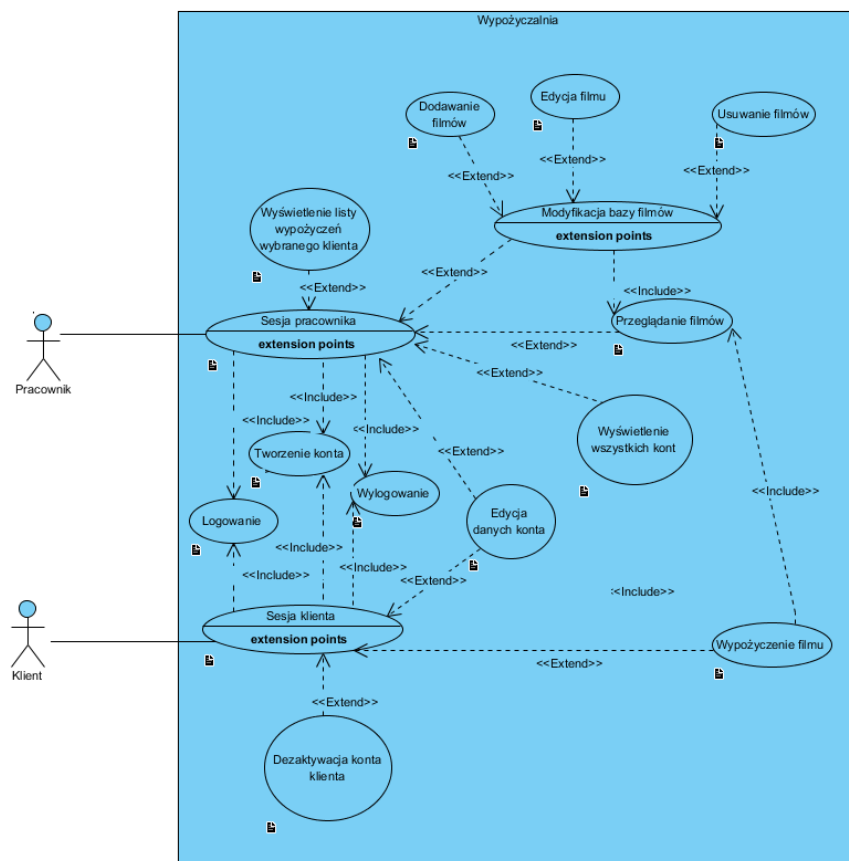
Data zakończenia:

Znajdź

Wróć do menu

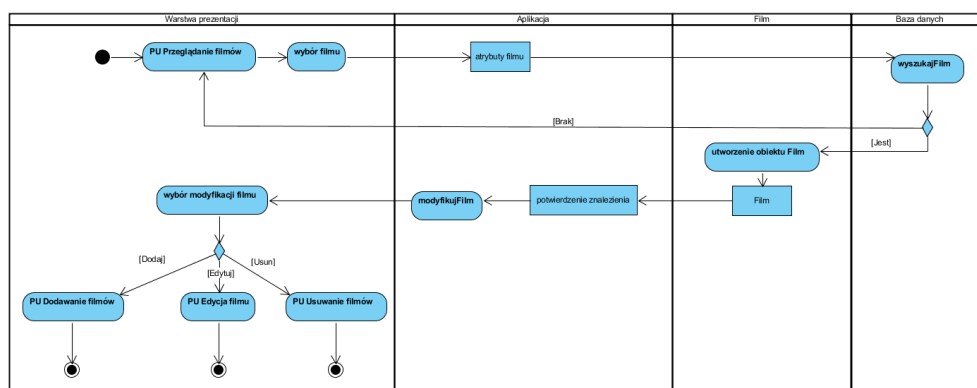
Rysunek 7: Okno do przeglądania wypożyczeń.

3.2.3 Diagram przypadków użycia



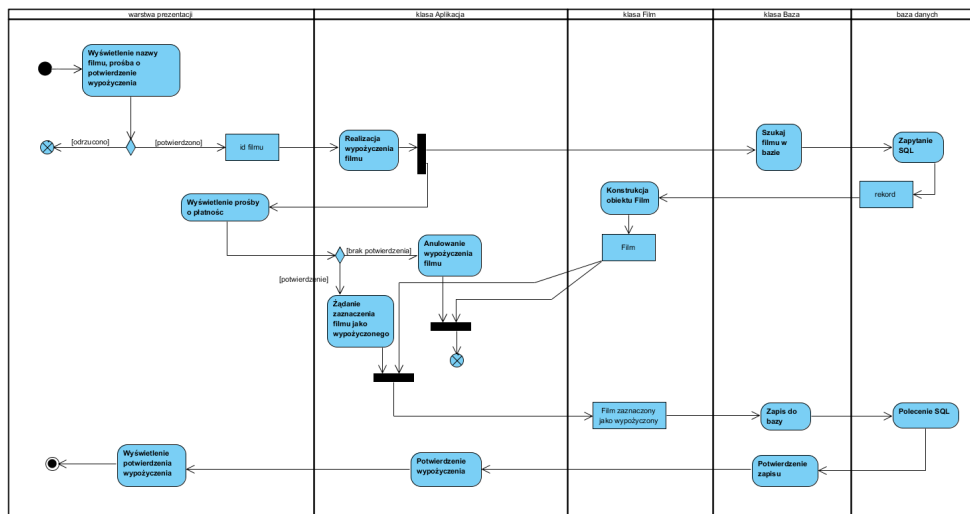
3.2.4 Wybrane diagramy czynności

Modyfikacja filmów



Użytkownik widzi listę filmów z PU Przeglądanie filmów. Po wybraniu filmu, jego atrybuty są przekazywane do zapytania bazy danych wyszukajFilm. Jeśli film nie istnieje w bazie, użytkownik z powrotem widzi listę filmów. Jeśli film istnieje, tworzymy obiekt Film, potwierdzamy znalezienie filmu i wywołujemy metodę modyfikujFilm. Użytkownik ma możliwość wyboru jednej z trzech modyfikacji filmu: dodaj, edytuj oraz usuń, dla których zachodzą odpowiednio PU: Dodawanie filmów, Edycja filmu i Usuwanie filmów.

Wypożyczenie filmu przez klienta



Najpierw program pyta klienta, czy na pewno chce wypożyczyć film. Jeśli klient odpowie przecząco, następuje koniec przypadku użycia. Następnie id filmu zostaje przekazane klasie Aplikacja w celu realizacji wypożyczenia filmu. Aplikacja rozpoczyna dwie czynności: prośbę o płatność i wyszukanie filmu w bazie do zaktualizowania jego statusu.

Szukanie filmu w bazie jest realizowane przez klasę Baza, która generuje odpowiednie zapytanie SQL. Baza danych odpowiada odpowiednim rekordem, który posłuży do utworzenia obiektu klasy Film.

W międzyczasie płatność zostaje potwierdzona (jeśli nie nastąpiło potwierdzenie, wypożyczenie zostaje anulowane). Dalej film zostaje zaznaczony jako wypożyczony i zaktualizowany w bazie. Baza zwraca potwierdzenie aktualizacji i aplikacja wyświetla odpowiedni komunikat dla użytkownika.

3.2.5 Metoda podłączania do bazy danych – integracja z bazą danych

Aplikacja będzie pisana w języku Java. Wybrany interfejsem do podłączenia do bazy danych jest Java DataBase Connectivity (JDBC). Interfejs ten jest bardzo często wykorzystywany w podobnych zastosowaniach.

3.2.6 Zabezpieczenia na poziomie aplikacji

Do zalogowania na konto klienta oraz pracownika potrzebne jest podanie hasła. Hasło musi być minimalnie 8-znakowe, z przynajmniej jedną wielką literą, jedną małą literą oraz przynajmniej jedną cyfrą. Hasło jest także potrzebne do usunięcia (dezaktywacji) konta klienta przez klienta. Konto pracownika nie może być utworzone w aplikacji, by uniemożliwić to klientowi. Będzie ono tworzone przez administratora bazy danych. Po wejściu do menu pracownika wyświetlone zostanie także ostrzeżenie, że wstęp jest zastrzeżony dla pracowników.

4 Implementacja systemu baz danych

Baza danych została zaimplementowana w MS SQL, za pomocą programu Microsoft SQL Server Management Studio.

Użycie w bazie collation: Polish_100_CI_AS_SC.

4.1 Tworzenie tabel i definiowanie ograniczeń

- CREATE DATABASE Wypożyczalnia
- CREATE TABLE Reżyserzy
(idR integer PRIMARY KEY NOT NULL,
imieR nvarchar(30) NOT NULL,
nazwiskoR nvarchar(60) NOT NULL)
- CREATE TABLE Aktorzy
(idA integer PRIMARY KEY NOT NULL,
imieA nvarchar(30) NOT NULL,
nazwiskoA nvarchar(60) NOT NULL)
- CREATE TABLE Gatunki
(nazwaG nvarchar(50) PRIMARY KEY NOT NULL)
- CREATE TABLE Filmy
(idF integer PRIMARY KEY NOT NULL,
nazwaF nvarchar(50) NOT NULL,
cenaF real NOT NULL,
rokPowstaniaF integer NOT NULL,
opisF nvarchar(MAX) NOT NULL)
- CREATE TABLE Wojewodztwa
(nazwaW nvarchar(19) PRIMARY KEY NOT NULL)
- CREATE TABLE AdresyKorespondencyjne
(idAD integer PRIMARY KEY NOT NULL,
miastoAD nvarchar(30) NOT NULL,
ulicaAD nvarchar(40) NULL,
numerDomuAD integer NOT NULL,
numerMieszkaniaAD integer NOT NULL,
nazwaW nvarchar(19) FOREIGN KEY REFERENCES Wojewodztwa(nazwaW))

- CREATE TABLE Uzytkownicy
(idU integer PRIMARY KEY NOT NULL,
nazwaU nvarchar(50) NOT NULL UNIQUE,
hasloU integer NOT NULL,
imieU nvarchar(30) NOT NULL,
nazwiskoU nvarchar(60) NOT NULL,
idAD integer FOREIGN KEY REFERENCES AdresyKorespondencyjne(idAD))
- CREATE TABLE Pracownicy
(idP integer PRIMARY KEY NOT NULL,
zarobkiP real NOT NULL,
dataZatrudnieniaP date NOT NULL,
idU integer FOREIGN KEY REFERENCES Uzytkownicy(idU))
- CREATE TABLE Wypozyczenia
(idW integer PRIMARY KEY NOT NULL,
cenaW real NOT NULL,
dataWypozyczeniaW date NOT NULL,
dataZwrotuW date NOT NULL,
idU integer FOREIGN KEY REFERENCES Uzytkownicy(idU),
idF integer FOREIGN KEY REFERENCES Filmy(idF))
- CREATE TABLE Filmy_Rezyserzy
(idF integer FOREIGN KEY REFERENCES Filmy(idF),
idR integer FOREIGN KEY REFERENCES Rezyserzy(idR))
- CREATE TABLE Filmy_Aktorzy
(idF integer FOREIGN KEY REFERENCES Filmy(idF),
idA integer FOREIGN KEY REFERENCES Aktorzy(idA))
- CREATE TABLE Filmy_Gatunki
(idF integer FOREIGN KEY REFERENCES Filmy(idF),
nazwaG nvarchar(50) FOREIGN KEY REFERENCES Gatunki(nazwaG))

4.2 Implementacja mechanizmów przetwarzania danych

4.2.1 Widoki

- CREATE VIEW WidokAdres AS
SELECT w.nazwaW, a.miastoAD, a.ulicaAD, a.numerDomuAD, a.numerMieszkaniaAD
FROM AdresyKorespondencyjne a JOIN Wojewodztwa w ON (w.nazwaW
= a.nazwaW);
- CREATE VIEW WidokFilm AS
SELECT f.nazwaF, f.cenaF, f.rokPowstaniaF, f.opisF, r.imieR, r.nazwiskoR,
a.imieA, a.nazwiskoA, g.nazwaG
FROM Filmy f JOIN Filmy_Rezyserzy rf ON (rf.idF = f.idF) JOIN
Rezyserzy r ON (rf.idR = r.idR)
JOIN Filmy_Aktorzy af ON (af.idF = f.idF) JOIN Aktorzy a ON
(af.idA = a.idA)
JOIN Filmy_Gatunki fg ON (fg.idF = f.idF) JOIN Gatunki g ON
(fg.nazwaG = g.nazwaG)

4.2.2 Indeksy

- CREATE UNIQUE INDEX INazwaU ON Uzytkownicy(nazwaU);
- CREATE INDEX INazwaF ON Filmy(nazwaF);
- CREATE INDEX IRokPowstaniaF ON Filmy(rokPowstaniaF);
- CREATE INDEX IDataWypozyczeniaW ON Wypozyczenia(dataWypozyczeniaW);

4.3 Implementacja uprawnień i innych zabezpieczeń

4.3.1 Uzytkownik

- CREATE LOGIN userAccount WITH PASSWORD = 'userAccount123';
- CREATE APPLICATION ROLE users
WITH PASSWORD = 'usersPassword192837465'
- USE Wypożyczalnia;
GRANT UPDATE ON Uzytkownicy TO users;
GRANT UPDATE ON AdresyKorespondencyjne TO users;

```

GRANT SELECT ON Filmy TO users;
GRANT SELECT ON Uzytkownicy TO users;
GRANT SELECT ON Wypozyczenia TO users;
GRANT SELECT ON Gatunki TO users;
GRANT SELECT ON Rezyserzy TO users;
GRANT SELECT ON Aktorzy TO users;
GRANT SELECT ON AdresyKorespondencyjne TO users;
GRANT SELECT ON Wojewodztwa TO users;
GRANT SELECT ON Filmy_Gatunki TO users;
GRANT SELECT ON Filmy_Aktorzy TO users;
GRANT SELECT ON Filmy_Rezyserzy TO users;
GRANT INSERT ON Uzytkownicy TO users;
GRANT INSERT ON Wypozyczenia TO users;
GRANT INSERT ON AdresyKorespondencyjne TO users;
GRANT SELECT ON WidokAdres TO users;
GRANT SELECT ON WidokFilm TO users;

```

4.3.2 Pracownik

- CREATE LOGIN workerAccount WITH PASSWORD = 'workerAccount77584';
- CREATE APPLICATION ROLE workers
WITH PASSWORD = 'workersPassword5847359'
- USE Wypożyczalnia;
GRANT UPDATE ON Filmy TO workers;
GRANT UPDATE ON Pracownicy TO workers;
GRANT UPDATE ON Gatunki TO workers;
GRANT UPDATE ON Rezyserzy TO workers;
GRANT UPDATE ON Aktorzy TO workers;
GRANT UPDATE ON AdresyKorespondencyjne TO workers;
GRANT SELECT ON Filmy TO workers;
GRANT SELECT ON Uzytkownicy TO workers;
GRANT SELECT ON Pracownicy TO workers;
GRANT SELECT ON Wypozyczenia TO workers;
GRANT SELECT ON Gatunki TO workers;
GRANT SELECT ON Rezyserzy TO workers;
GRANT SELECT ON Aktorzy TO workers;

```

GRANT SELECT ON AdresyKorespondencyjne TO workers;
GRANT SELECT ON Wojewodztwa TO workers;
GRANT SELECT ON Filmy_Gatunki TO workers;
GRANT SELECT ON Filmy_Aktorzy TO workers;
GRANT SELECT ON Filmy_Rezyserzy TO workers;
GRANT INSERT ON Filmy TO workers;
GRANT INSERT ON Uzytkownicy TO workers;
GRANT INSERT ON Pracownicy TO workers;
GRANT INSERT ON Gatunki TO workers;
GRANT INSERT ON Rezyserzy TO workers;
GRANT INSERT ON Aktorzy TO workers;
GRANT INSERT ON AdresyKorespondencyjne TO workers;
GRANT INSERT ON Filmy_Gatunki TO workers;
GRANT INSERT ON Filmy_Aktorzy TO workers;
GRANT INSERT ON Filmy_Rezyserzy TO workers;
GRANT SELECT ON WidokAdres TO workers;
GRANT SELECT ON WidokFilm TO workers;
GRANT UPDATE ON Filmy_Gatunki TO workers;
GRANT UPDATE ON Filmy_Aktorzy TO workers;
GRANT UPDATE ON Filmy_Rezyserzy TO workers;

```

Obu stworzonym loginom, trzeba przyznać w ustawieniach dostęp do stworzonej bazy danych Wypożyczalnia.

4.4 Testowanie bazy danych na przykładowych danych

4.4.1 Dodawanie prawidłowych danych i wygląd tabeli po dodaniu

Wszystkie dane zostały ukazane na 5 przykładach w celu zwiększenia czytelności sprawozdania. Do bazy danych zostało dodanych około 30 pozycji w każdej z tabel.

Rezyserzy

```

INSERT INTO Rezyserzy
VALUES (1,'Steven', 'Spielberg'),
(2,'Peter', 'Jackson'),
(3,'Martin', 'Scorsese'),

```

(4,'Christopher', 'Nolan'),
(5,'Steven', 'Soderbergh')

	idR	imieR	nazwiskoR
1	1	Steven	Spielberg
2	2	Peter	Jackson
3	3	Martin	Scorsese
4	4	Christopher	Nolan
5	5	Steven	Soderbergh
6	6	Ridley	Scott
7	7	Quentin	Tarantino

Aktorzy

```
INSERT INTO Aktorzy  
VALUES (1, 'Robert', 'De Niro'),  
(2, 'Al', 'Pacino'),  
(3, 'Kevin', 'Spacey'),  
(4, 'Jack', 'Nicholson'),  
(5, 'Sean', 'Connery')
```

	idA	imieA	nazwiskoA
1	1	Robert	De Niro
2	2	Al	Pacino
3	3	Kevin	Spacey
4	4	Jack	Nicholson
5	5	Sean	Connery
6	6	Harrison	Ford
7	7	Russell	Crowe

Gatunki

```
INSERT INTO Gatunki  
VALUES ('akcja'),  
( 'animacja'),  
( 'anime'),  
( 'biograficzny'),  
( 'dokumentalny')
```

	nazwaG
1	akcja
2	animacja
3	anime
4	biograficzny
5	dokumentalny
6	dramat
7	dramat historyczny
8	edukacyjny
9	etiuda
10	familiiny

Filmy

INSERT INTO Filmy

VALUES (1,'E.T.', 5, 1985,'Pozostawiony przypadkowo na Ziemi kosmita zaprzyjaźnia się z małym chłopcem, Elliotem. Niezwykły przybysz próbuje wrócić do domu.'),

(2,'Lista Schindlera', 5, 1992,'Historia przedsiębiorcy Oskara Schindlera, który podczas II wojny światowej uratował przed pobytem w obozach koncentracyjnych 1100 Żydów.'),

(3,'Władca Pierścieni: Drużyna pierścienia', 15, 2007,'Podróż hobbita z Shire i jego ośmiu towarzyszy, której celem jest zniszczenie potężnego pierścienia pożądanego przez Czarnego Władcę - Saurona.'),

(4,'King Kong', 15, 1989,'Lata 30. XX wieku. Bankrutujący reżyser i głodująca aktorka wyruszają na tajemniczą Wyspę Czaszki, by nagrać film swojego życia, ponieważ w Nowym Jorku panuje wielki kryzys.'),

(5,'Ulice nędzy', 10, 1993,'Charlie pracuje dla mafii kontrolującej nowojorską Małą Italię. Na drodze do upragnionego awansu w przestępczej hierarchii staje mu przyjaźń z zadłużonym cwaniaczkiem Johnnym Boyem.')

	idF	nazwaF	cenaF	rokPowstaniaF	opisF
1	1	E.T.	5	1985	Pozostawiony przypadkowo na Ziemi kosmita zaprzyj...
2	2	Lista Schindlera	5	1992	Historia przedsiębiorcy Oskara Schindlera, który pod...
3	3	Władca Pierścieni: Drużyna pierścienia	15	2007	Podróż hobbita z Shire i jego ośmiu towarzyszy, której...
4	4	King Kong	15	1989	Lata 30. XX wieku. Bankrutujący reżyser i głodująca ...
5	5	Ulice nędzy	10	1993	Charlie pracuje dla mafii kontrolującej nowojorska Ma...
6	6	Chłopcy z ferajny	10	1986	Kilkunastoletni Henry i Tommy DeVito trafiają pod opi...
7	7	Memento	12,5	1985	Leonard, szukający mordercy swojej żony, w wyniku ...
8	8	Mroczny Rycerz	14	1986	Batman, z pomocą porucznika Gordona oraz prokura...
9	9	Co z oczu, to z serca	6	1985	Jack Foley ucieka z więzienia. Do pościgu zostaje w...
10	10	Traffic	7	2005	Trzy historie opowiadające o walce z narkotykami prz...

Filmy__Aktorzy

```
INSERT INTO Filmy__Aktorzy  
VALUES (1, 4),  
(1, 15),  
(1, 23),  
(2, 1),  
(2, 7)
```

	idF	idA
1	1	4
2	1	15
3	1	23
4	2	1
5	2	7
6	2	9

Filmy__Rezyserzy

```
INSERT INTO Filmy__Rezyserzy  
VALUES (1, 16),  
(2, 13),  
(2, 21),  
(3, 1),  
(4, 16)
```

	idF	idR
1	1	16
2	2	13
3	2	21
4	3	1
5	4	16
6	2	9

Filmy__Gatunki

```
INSERT INTO Filmy__Gatunki  
VALUES (1,'biograficzny'),  
(1,'komedia'),  
(1,'przygodowy'),  
(2,'akcja'),  
(2,'dramat')
```

	idF	nazwaG
1	1	biograficzny
2	9	kryminał
3	1	komedia
4	1	przygodowy
5	2	akcja
6	2	dramat

Wojewodztwa

```
INSERT INTO Wojewodztwa  
VALUES ('pomorskie'),  
( 'zachodnio-pomorskie'),  
( 'lubuskie'),  
( 'dolnośląskie'),  
( 'opolskie')
```

	nazwaW
1	dolnośląskie
2	kujawsko-pomorskie
3	lubelskie
4	lubuskie
5	łódzkie
6	małopolskie
7	mazowieckie
8	opolskie
9	podkarpackie

Adresy korespondencyjne

```
INSERT INTO AdresyKorespondencyjne
VALUES (1,'Wrocław','Polna',30,10,'dolnośląskie'),
(2,'Opole','Leśna',29,9,'opolskie'),
(3,'Warszawa','Słoneczna',24,14,'mazowieckie'),
(4,'Kraków','Krótka',22,12,'małopolskie'),
(5,'Brzeg','Szkolna',21,15,'opolskie')
```

	idAD	miastoAD	ulicaAD	numerDomuAD	numerMieszkaniaAD	nazwaW
1	1	Wrocław	Polna	30	10	dolnośląskie
2	2	Opole	Leśna	29	9	opolskie
3	3	Warszawa	Słoneczna	24	14	mazowieckie
4	4	Kraków	Krótka	22	12	małopolskie
5	5	Brzeg	Szkolna	21	15	opolskie
6	6	Wrocław	Ogrodowa	21	742	dolnośląskie

Uzytkownicy

```
INSERT INTO Uzytkownicy
VALUES (1,'Pracownik1213', 2010727,'Mateusz','Sikorski',1),
(2,'Pracownik1214', 5285819,'Beata','Klimczyk',2),
(3,'Pracownik1215', 1252624,'Jan','Morykło',3),
(4,'Karolek123', 246715,'Karol','Pożonko',4),
(5,'AgataP604', 113242,'Agata','Jarząbek',5)
```

	idU	nazwaU	hasloU	imieU	nazwiskoU	idAD
1	1	Pracownik1213	2010727	Mateusz	Sikorski	1
2	2	Pracownik1214	5285819	Beata	Klimczyk	2
3	3	Pracownik1215	1252624	Jan	Morykło	3
4	4	Karolek123	246715	Karol	Pożonko	4
5	5	AgataP604	113242	Agata	Jarząbek	5
6	6	Baziol411	2451715	Bazio	Motyka	6
7	7	Berfs	2452615	Piotr	Wieczorek	7

Pracownicy

```
INSERT INTO Pracownicy  
VALUES (1,2500,convert(datetime,'10-05-15',5),1),  
(2,2550,convert(datetime,'21-08-14',5),2),  
(3,2600,convert(datetime,'30-11-16',5),3)
```

	idUP	zarobkiP	dataZatrudnieniaP	idU
1	1	2500	2015-05-10	1
2	2	2550	2014-08-21	2
3	3	2600	2016-11-30	3

Wypożyczenia

```
INSERT INTO Wypozyczenia  
VALUES (1,14,convert(datetime,'10-05-15',5),convert(datetime,'17-05-15',5),5,12),  
(2,15,convert(datetime,'11-05-15',5),convert(datetime,'18-05-15',5),7,33),  
(3,8,convert(datetime,'15-05-15',5),convert(datetime,'22-05-15',5),19,30),  
(4,15,convert(datetime,'30-06-15',5),convert(datetime,'07-07-15',5),14,2),  
(5,18,convert(datetime,'02-07-15',5),convert(datetime,'09-07-15',5),5,21)
```

	idW	cenaW	dataWypozyczeniaW	dataZwrotuW	idU	idF
1	1	14	2015-05-10	2015-05-17	5	12
2	2	15	2015-05-11	2015-05-18	7	33
3	3	8	2015-05-15	2015-05-22	19	30
4	4	15	2015-06-30	2015-07-07	14	2
5	5	18	2015-07-02	2015-07-09	5	21
6	6	13	2015-07-02	2015-07-09	6	25

4.4.2 Przykładowe dodawanie błędnych danych

Przekroczenie zakresu nvarchar

Komenda:

```
INSERT INTO AdresyKorespondencyjne  
VALUES (101,'Wrocławwwwwwwwwwwwwwwwwwwwwwwww', 'Polna',30,10,'dolnośląskie')
```

Błąd:

String or binary data would be truncated.
The statement has been terminated.

Opis błędu:

Przekroczenie zakresu pola miastoAD (31, gdzie maksymalny to 30).

Próba ponownego dodania istniejącej wartości unikatowej (klucza głównego)

Komenda:

```
INSERT INTO AdresyKorespondencyjne  
VALUES (1,'Wrocław', 'Polna',30,10,'dolnośląskie')
```

Błąd:

Violation of PRIMARY KEY constraint 'PK__AdresyKo__9DB8A6A01E58FB0C'.
Cannot insert duplicate key in object 'dbo.AdresyKorespondencyjne'. The
duplicate key value is (1).
The statement has been terminated.

Opis błędu:

Próba ustawienia w polu idAD istniejącej już wartości 1.

Błędne dodanie typu nvarchar

Komenda:

```
INSERT INTO AdresyKorespondencyjne  
VALUES (100,Wrocław,'Polna',30,10,'dolnośląskie')
```

Błąd:

Invalid column name 'Wrocław'.

Opis błędu:

Dodanie nvarchar bez apostrofów.

Przekroczenie wartości integer

Komenda:

```
INSERT INTO AdresyKorespondencyjne  
VALUES (100,'Wrocław','Polna',300000000000,10,'dolnośląskie')
```

Błąd:

Arithmetic overflow error converting expression to data type int.
The statement has been terminated.

Opis błędu:

Podanie wartości integer przekraczającej zakres 4B.

Nieprawidłowa ilość argumentów przy niejawnym dodawaniu danych

Komenda:

```
INSERT INTO AdresyKorespondencyjne  
VALUES (100,'Wrocław','Polna',30,10)
```

Błąd:

Column name or number of supplied values does not match table definition.

Opis błędu:

Nie podano ostatniego argumentu - nazwy województwa.

4.4.3 Sprawdzenie poprawności działania widoków

Widok adresu

Komenda:

```
SELECT * FROM WidokAdres  
WHERE miastoAD='Opole'
```

Wynik:

	nazwaW	miastoAD	ulicaAD	numerDomuAD	numerMieszkaniaAD
1	opolskie	Opole	Leśna	29	9
2	opolskie	Opole	Lipowa	168	84
3	opolskie	Opole	Brzozowa	157	52
4	opolskie	Opole	Brzozowa	156	52
5	opolskie	Opole	Łąkowa	155	108
6	opolskie	Opole	Łąkowa	154	208

Widok filmu

Komenda:

```
SELECT * FROM WidokFilm  
WHERE nazwiskoA='Walken'
```

Wynik:

	nazwaF	cenaF	rokPowstaniaF	opisF	imieR	nazwiskoR	imieA	nazwiskoA	nazwaG
1	Mroczny Rycerz	14	1986	Batman, z pomocą p...	Ron	Howard	Christopher	Walken	thriller
2	Sweeney Todd:...	13	1999	Golibroda wraca do ...	Ron	Howard	Christopher	Walken	thriller

4.4.4 Sprawdzenie uprawnień konta użytkownika korzystającego z roli aplikacji użytkownika

Po zalogowaniu do bazy danych za pomocą danych do logowania dla użytkownika userAccount nie mamy jeszcze dostępu do tabel. Przedstawia to następujący przykład:

Komenda:

```
USE Wypożyczalnia  
SELECT * FROM Filmy
```

Błąd:

The SELECT permission was denied on the object 'Filmy', database 'Wypożyczalnia', schema 'dbo'.

Przyznanie użytkownikowi dostępu do stworzonej roli aplikacji „users” następuje przy pomocy następującej komendy:

```
EXEC sp_setapprole 'users', 'usersPassword192837465'
```

Po przyznaniu roli aplikacji działają wszystkie tabele do których dostęp został przyznany. W przypadku konta pracownika jedyna różnica to nazwa roli aplikacji oraz hasło.

5 Implementacja i testy aplikacji

W tej części przedstawiona zostanie zaimplementowana aplikacja łącząca się ze stworzoną wcześniej bazą danych. Kolejno przedstawiony zostanie sposób instalacji aplikacji, instrukcja jej użytkowania, testowanie wybranych funkcji oraz omówienie wybranych rozwiązań programistycznych.

5.1 Instalacja i konfigurowanie systemu

Poniższa instrukcja instalacji, konfiguracji oraz używania aplikacji zawarta jest także w pliku `README.TXT` dołączanym wraz z aplikacją. Stworzona aplikacja desktopowa zawarta jest w pliku wykonywalnym `Wypożyczalnia.jar`. Na jej otwarcie niezbędne będzie zainstalowanie jre (ang. *Java Runtime Environment*), do którego instalator załączony został w folderze z aplikacją pod nazwą `JavaSetup8u151.exe`. W folderze `lib` zawarte są biblioteki używane przez aplikację. Żeby utworzyć bazę danych niezbędne jest otwarcie dołączonego z aplikacją skryptu `Wypożyczalnia.sql` za pomocą programu **Microsoft SQL Server Management Server 17** na porcie 1433.

5.2 Instrukcja użytkowania aplikacji

Aplikacja stworzona została zarówno dla klientów wypożyczalni jak i dla sprzedawców. Po otwarciu aplikacji możliwy jest wybór menu klienta, menu sprzedawcy lub wyjście z aplikacji.

Menu klienta

W menu klienta możliwe jest stworzenie konta przez podanie: nazwy konta, hasła, imienia, nazwiska, województwa, miasta, ulicy (nieobowiązkowa), numeru domu oraz numeru mieszkania (nieobowiązkowe). Wymaganie dotyczące hasła jest takie, że musi mieć od 8 do 16 znaków. Możliwe jest także zalogowanie przy pomocy podania nazwy konta oraz hasła, oraz powrót do menu głównego aplikacji.

Po zalogowaniu się na konto widać dane zalogowanego użytkownika oraz pierwszą stronę listy filmów (5 różnych filmów). Mamy możliwość zmiany hasła, adresu, a także nazwiska. Możemy usunąć konto (dezaktywować) oraz przejrzeć historię wypożyczeń dla naszego konta. W opcjach konta jest także przeglądanie listy filmów. Za pomocą guzików do zmiany stron zmieniają się

nazwy filmów na 5 różnych guzikach. Możliwe jest wyszukanie filmów przez podanie nazwy filmu, gatunku filmu lub roku powstania filmu. Po wybraniu filmu przechodzimy na stronę wypożyczenia na której widzimy następujące dane wybranego filmu: nazwa filmu, gatunek, rok powstania, opis filmu, cena wypożyczenia oraz główni reżyserzy i aktorzy. W tym momencie możliwe jest przejście do płatności a następnie przejście do przelewu bankowego lub powrót do konta.

Menu sprzedawcy

Przy wejściu do menu sprzedawcy otrzymujemy informację ostrzegającą, że nieautoryzowane wejście może być ukarane. Jest to informacja dla klientów, którzy próbowaliby zalogować się jako sprzedawca. W menu sprzedawcy mamy jedynie możliwość zalogowania się. Administrator bazy danych musi stworzyć konto pracownika. Po zalogowaniu się na konto widać dane zalogowanego pracownika. Możliwa jest zmiana hasła, nazwiska oraz adresu. Możliwe jest wyświetlenie wszystkich kont utworzonych w bazie danych, przejrzanie wypożyczeń oraz edycja filmów, a także wylogowanie się z konta.

Po wybraniu przeglądania wypożyczeń otrzymujemy stronę ze wszystkimi wypożyczeniami które możemy przeglądać przy pomocy guzików "Poprzednie wypożyczenie" oraz "Następne wypożyczenie". Wypożyczenia możemy wyszukać podając: ID wypożyczenia, nazwę filmu lub nazwę klienta.

Po wybraniu edycji filmów otrzymujemy listę filmów, podobną jak w przypadku menu klienta, pola które zawierać będą informacje o: ID filmu, nazwie filmu, gatunku filmu, roku powstania, opisie filmu, aktualnej cenie oraz najważniejszych reżyserach i aktorach. Mamy także opcje wyszukania, edycji, utworzenia oraz usunięcia filmu. Przeglądanie filmów polega na zmianie stron przy pomocy przycisków "Poprzednia strona" oraz "Następna strona". Po wybraniu filmu, pola wypełniają się informacjami o wybranym filmie. Wyszukiwanie filmu (guzik "Znajdź") pobiera jedynie wpisaną nazwę filmu, gatunek (koniecznie jeden) oraz rok powstania filmu i zwraca wyniki na liście filmów. Edycja filmu wyszukuje podany film i przypisuje do niego zmienione dane (pola nie mogą być puste). Do usunięcia filmu (dezaktywacji) konieczne jest potwierdzenie, żeby film nie został usunięty przez przypadek.

5.3 Testowanie opracowanych funkcji systemu

Testy jednostkowe i akceptacyjne nastąpiły przy pomocy narzędzi JUnit, JMockit oraz FitNess. Testowane były głównie funkcje klasy Wypożyczalnia, która zawiera funkcję łączącą się z bazą danych, funkcje realizujące połączenie do ról aplikacji, oraz funkcje realizujące zapytania na bazie danych (np. znajduje film o podanej nazwie, znajduje konto użytkownika o podanej nazwie i hasle).

5.4 Omówienie wybranych rozwiązań programistycznych

5.4.1 Implementacja interfejsu dostępu do bazy danych

Dostęp do bazy danych odbywał się poprzez interfejs jdbc (ang. *Java Database Connectivity*). Wszystkie funkcje komunikujące się z bazą danych znajdują się w klasie Wypożyczalnia. Interfejs ten jest użyty, ponieważ jest bardzo często wykorzystywany w podobnych zastosowaniach. Przykłady funkcji komunikujących się z bazą danych:

Nawiązanie połączenia z bazą danych

```
void initializeConnection(String dbConnect, String user, String password)
throws IOException, InterruptedException, ClassNotFoundException,
SQLException{
    DriverManager.registerDriver(new SQLServerDriver());
    con = DriverManager.getConnection(dbConnect, user, password);
}
```

Przydzielenie roli aplikacji użytkownika

```
void userRole() throws SQLException{
    Statement stmt2 = con.createStatement();
    String SQL = "USE Wypożyczalnia EXEC sp_setapprole 'users',
'usersPassword192837465' SELECT * from Aktorzy";
    rs = stmt2.executeQuery( SQL );
    rs.close();
    stmt2.close();
}
```

Ustawienie podanego nazwiska do konta o podanej nazwie

```
void setNazwisko(String nazwisko, String nazwa) throws SQLException{
    stmt = con.prepareStatement("SELECT * FROM Uzytkownicy WHERE nazwaU
        = ?", ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
    stmt.setString(1, nazwa);
    rs = stmt.executeQuery();
    rs.first();
    rs.updateString("nazwiskoU", nazwisko);
    rs.updateRow();
    stmt.close();
    rs.close();
    return;
}
```

5.4.2 Implementacja wybranych funkcjonalności systemu

Większość klas odpowiada za pojedyncze strony widoczne w aplikacji. Są to między innymi klasy: MenuGlowne, MenuSprzedawca, czy WypozyczFilm. Oprócz nich jest klasa Wypozyczalnia zawierająca wszystkie funkcje komunikujące się z bazą danych, a także klasy pomocnicze zawierające informacje o filmach, aktorach, reżyserach czy wypożyczeniach nazywające się odpowiednio: Aktor, Film czy Wypożyczenie. Klasy pomocnicze pomagają uporządkować dane w przypadku zwracania większej ilości informacji przez funkcje klasy Wypozyczalnia.

5.4.3 Implementacja mechanizmów bezpieczeństwa

- ostrzeżenie o nieautoryzowanym wejściu do menu sprzedawcy,
- hashowanie haseł w bazie danych za pomocą funkcji hashCode(),
- hashowanie haseł przy ich wpisywaniu w odpowiednich polach,
- hasło musi mieć od 8 do 16 znaków,
- przy usuwaniu konta oraz filmu niezbędne jest potwierdzenie.

6 Podsumowanie i wnioski

W projekcie wypożyczalni filmów online przedstawiono analizę wymagań. Następnie ułożony został projekt bazy danych oraz aplikacji desktopowej dla użytkowników i pracowników. Następnie zaimplementowane i opisane zostały kolejno: baza danych oraz aplikacja.

Baza danych, aplikacja oraz połączenie działają bez żadnych błędów. Przy opisie aplikacji przedstawione zostały niezbędne kroki do instalacji aplikacji oraz instrukcja użytkowania aplikacji. Wszystkie niezbędne pliki zostały dołączone wraz z powyższym sprawozdaniem. Jeżeli pliki nie zostały dołączone, można je znaleźć wraz z aplikacją pod poniższym adresem:

<https://github.com/AdamFilipowicz/WypozyczalniaFilmowOnline>.

Literatura

[1] <https://www.microsoft.com/pl-pl/sql-server/sql-server-2016>

[2] <https://netbeans.org/>

[3] <https://docs.oracle.com/javase/8/docs/>