**Aplikacja do zarządzania kinem**

Wykonał:

Paweł Lesiak

Spis treści

[1.Opis aplikacji. 2](#_Toc102994998)

[2. Reguły biznesowe. 2](#_Toc102994999)

[3. Struktura bazy danych. 2](#_Toc102995000)

[5. Kod 3](#_Toc102995001)

[6. Screeny z aplikacji 3](#_Toc102995002)

# 1.Ważne informacje.

**Dla użycia aplikacji utworzyłem dwa konta:**

Administrator:

Hasło: admin

Login: admin

Użytkownik:

Hasło: user

Login: user

# 2.Opis aplikacji.

Aplikacja desktopowa wykonana w języku C# oraz przy użyciu MySQL workbench, została wykonana do obsługi kina. Posiada możliwości logowania dla administratora – właściciela kina oraz pracownika.

Dla administratora aplikacja zapewnia możliwości:

- ustawiania seansów,

- dodawania, usuwania i edytowania filmy

- zatrudniania, edytowania i zwalniania pracownika

- podglądu na bilans rachunków (zysk, koszt, ogółem)

Dla pracownika:

- w momencie zalogowania włączany jest zegar który odmierza sesje zalogowania, na tej podstawie jest wyliczany jego zarobek.

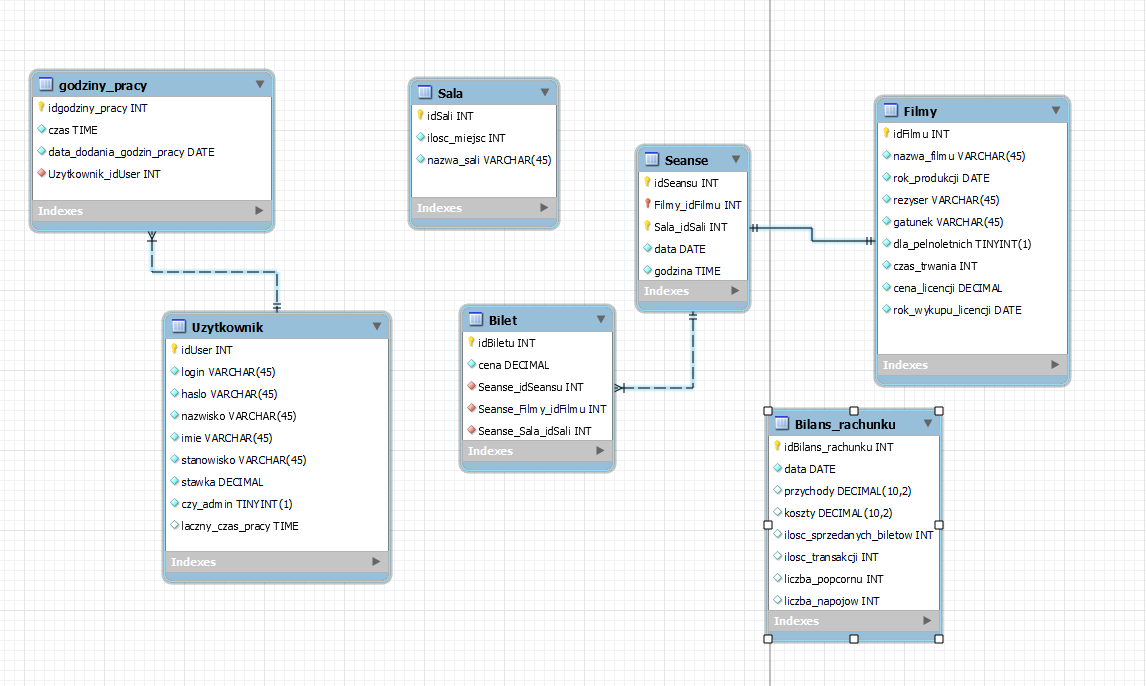
- sprzedaży biletów, napojów, popcornu

- podgląd swoich logowań oraz zarobku

# 3. Reguły biznesowe.

Przy zakupie 2 biletów – popcorn gratis, przy zakupie 6 biletów – 20% rabat na wszystko

# 4. Struktura bazy danych.



**Relacje pomiędzy tabelami:**

Godziny\_pracy – użytkownik – jeden do wielu – (użytkownik może mieć wiele godzin pracy)

Bilet – Seanse – jeden do wielu – (seans może mieć wiele biletów)

Seanse – Filmy – jeden do jednego (każdy seans może mieć jeden film)

# 5. Kod

Utworzyłem specjalną klasę **ConnectionDB** w której znajdują się wszystkie połączenia z bazą danych.

W klasie ten znajduje się inicjalizująca połączenie z bazą danych:  
  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Oraz zamknięcie i otwarcie połączenia. Metody te wykorzystywane są przy każdym połączeniu dzięki wywołaniu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Poniżej znajdują się nazwy metod i ich zastosowanie:

**CheckLogin**(string login, string password) – do sprawdzenia poprawności logowania

**CheckTypeUser**(int id) – do sprawdzenia typu użytkownika

**UserById**(int id) – do stworzenia obiektu user na podstawie id

**GetDataGridView**(string command, DataGridView datagridView) – najczęściej używana metoda. Wykorzystywana do obsługi data grid view. Zawiera obiekty BindingSourcem MySqlDataAdapter oraz DataTable.

**SendIncome**(string currentData, int numberTransaction, decimal income = 0, int numberTicketsSold = 0, int numberPopcorns = 0, int numberDrinks = 0) – do przesłania przychodu po wylogowaniu

**SendCost**(string currentData, double dailyWage) – do wysłania kosztów utrzymania pracownika

**RegisterWorkingHours**(int id, string workTimer, string currentDate) – do zarejestrowania godzin pracy użytkownika

**GetRegisterWorkingHours**(int id) – do otrzymania informacji na temat przepracowanych godzin

**UpdateRegisterWorkingHours**(int id, string totalWorktime) – do zaktualizowania czasu rpacy pracownika

**AddNewEmployee**(string login, string password, string lastname, string name, string position, double rate) – dodanie nowego pracownika

**RemoveEmployee**(int id) – usunięcie pracownika

**EditEmployee**(int id, string login, string haslo, string lastname, string name, string position, decimal rate) – do edycji pracownika

**AddNewMovie**(string title, string productionDate, string director, string species, int forAdults, double duration, double licencePrice, string datePurchase) – do dodania nowego filmu

**RemoveMovie**(int id) – do usunięcia filmu

**GetMoviesToDashboard**(string date) – do otrzymania listy filmów granych w danym dniu

**GetHoursMovieToDashbord**(string movies,string date) – do otrzymania godzin granego filmu

**getInfoAboutMovie**(string movie, string snapshot) – do otrzymania informacji na temat filmu

**GetInfoAboutHalls**(string movie, string hour, string day) – otrzymanie ifnroamcji na temat Sali

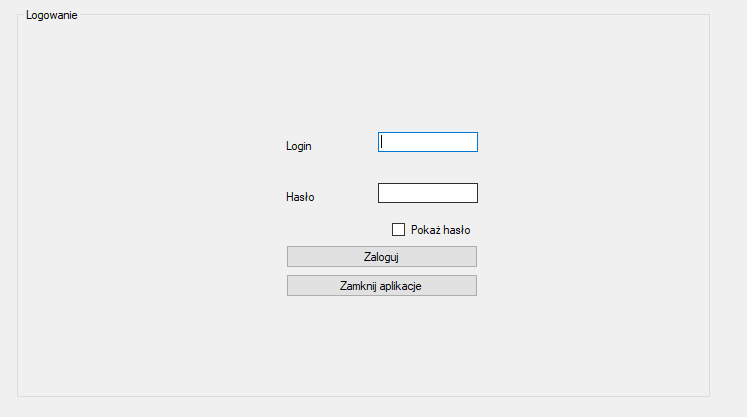
**GetIdsSeans**(string movie, string hall, string day, string hour) – do pobrania id seansu

**AddTicketsToDB**(List<int> ids) – dodanie biletów

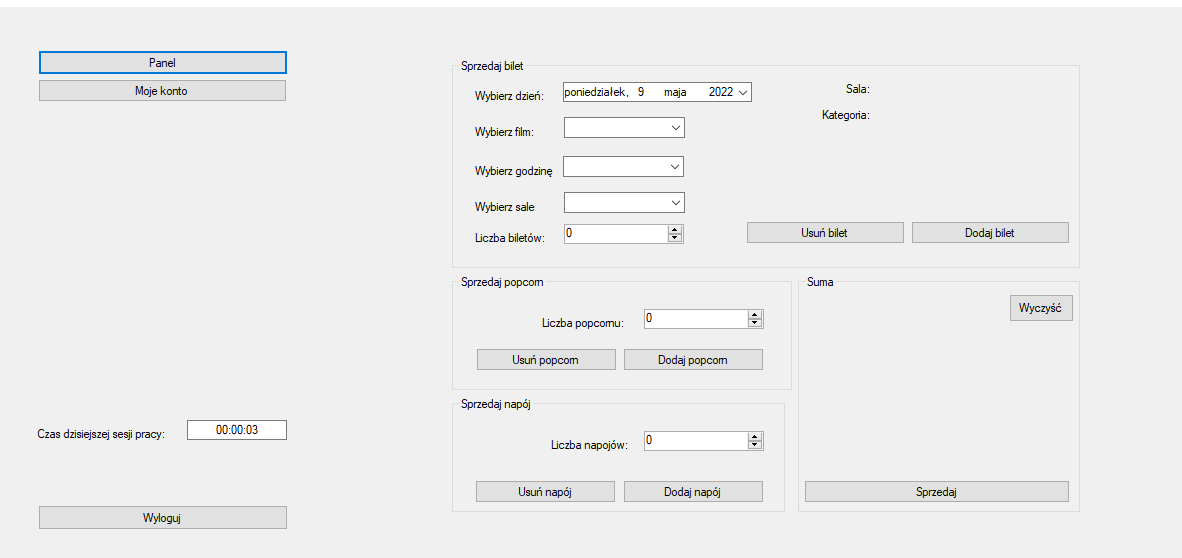
**AddNewSeans**(int idMovie, int idHall, string date, string hour) – dodanie nowego seansu

**RemoveSeans**(int idSeans, int idMovie, int idHall) – usunięcie seansu

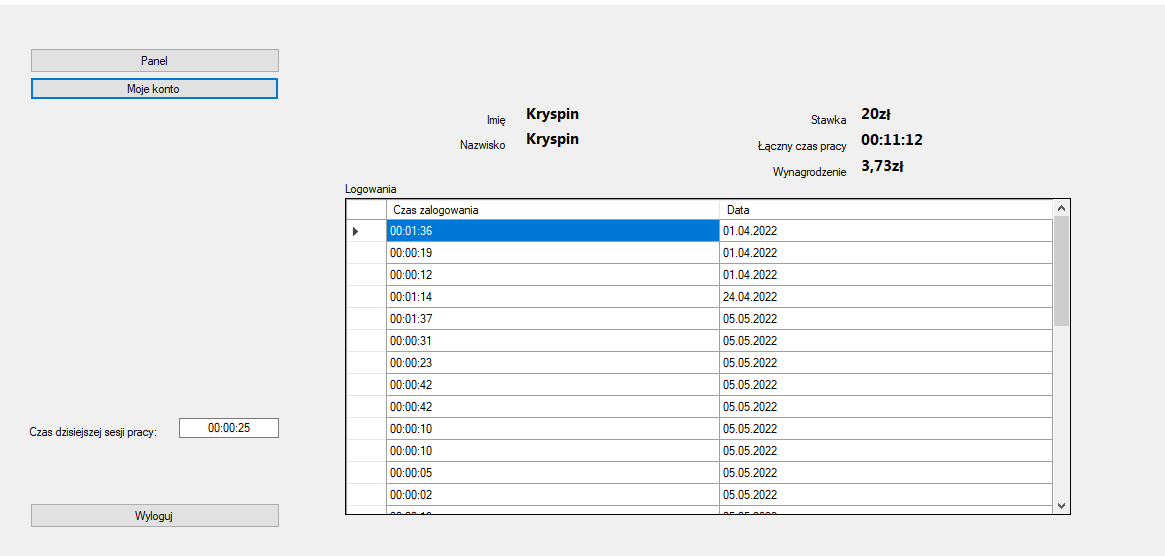
# 6. Screeny z aplikacji

Panel logowania umożliwia dostanie się do aplikacji poprzez wprowadzenie prawidłowego loginu oraz hasła. Dzięki checkboxowi możliwe jest włączenie/wyłączenie widoczności hasła. 

Użytkownik – panel pracownika – umożliwia sprzedaż biletów na podstawie wybranego dnia, filmu, godziny, sali, a także popcornu oraz napojów. Przez cały czas zalogowania użytkownika działa licznik który nalicza czas pracy pracownika. Przycisk sprzedaj dodaje dane do bazy danych. Przycisk wyloguj dodaje cały przychód z dnia oraz koszt utrzymania pracownika.



Użytkownik – moje konto – w tym panelu użytkownik może zobaczyć swój czas pracy oraz swoje łączne wynagrodzenie.



Administrator – filmy – w tym panelu administrator może dodać, edytować i usuwać filmy.

Obraz zawierający tekst, zrzut ekranu, wewnątrz

Opis wygenerowany automatycznie

Administrator – pracownicy - w tym panelu administrator może dodać, edytować i zwalniać pracowników.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Administrator – raport sprzedaży – w tym panelu administrator ma do wglądu swoje przychody. Może również skorzystać z kalendarza który będzie wskazywał przychód tylko z danego dnia.

Obraz zawierający stół

Opis wygenerowany automatycznie

Administrator – raport kosztów - w tym panelu administrator ma do wglądu swój koszt. Może również skorzystać z kalendarza który będzie wskazywał koszt tylko z danego dnia.

Obraz zawierający stół

Opis wygenerowany automatycznie

Administrator bilans: jest to zestawienie kosztów oraz przychodów.  
  
Obraz zawierający stół

Opis wygenerowany automatycznie

Administrator – seanse – w tym panelu użytkownik może dodawać seanse na podstawie id Sali, id filmu, wybranej daty oraz godziny. Może również usuwać wybrany seans.

Obraz zawierający stół

Opis wygenerowany automatycznie