

# **Multipleks kinowy**

## **Dokumentacja**



Wersja 1.2

Grupa habeja, wtorek 17:50, członkowie:

Krzysztof Hardek

Adam Bera

Grzegorz Janosz

# Opis projektu

Projekt polega na stworzeniu aplikacji realizującej multipleks kinowy. Aplikacja ma umożliwiać:

- Wprowadzanie osób, w tym z różnymi uprawnieniami,
- Dostęp do aktualnego repertuaru kina,
- System wysyła maile z powiadomieniami
- Statystyki, Wyszukiwanie, Polecanie filmu

Jego realizacja została podzielona na trzy kamienie milowe. Wykonanie każdego z nich skutkuje stworzeniem działającego prototypu

## M1

### 1. Planowany progres:

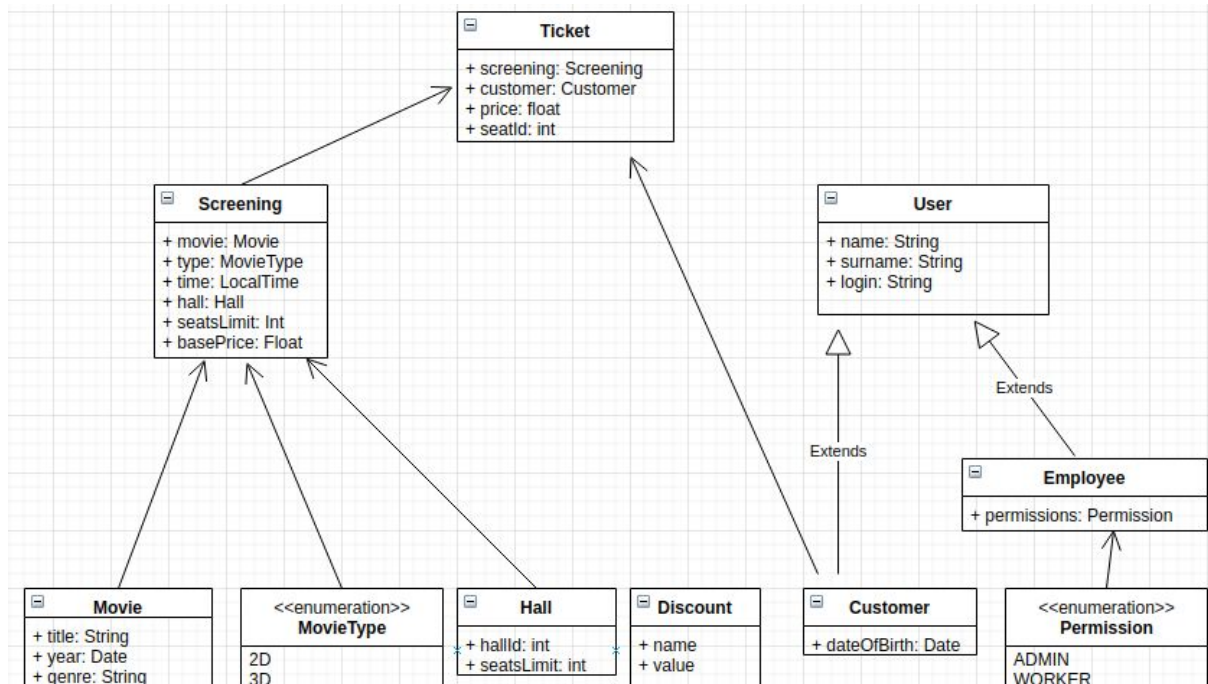
- kompletny model,
- wprowadzanie w aplikacji osób i sal kinowych

### 2. Podział obowiązków

- Adam Bera - GUI (dla osób i sal)
- Grzegorz Janosz - Baza danych i jej interfejs (dla osób i sal)
- Krzysztof Hardek - Przygotowanie dokumentacji, logika aplikacji (podpięcie bazy i GUI), podstawowe klasy z diagramu (implementacja)
- Cały zespół - Stworzenie modelu (praca koncepcyjna)

### 3. Co zostało zrobione

**Model danych:**



Stworzyliśmy model danych w postaci diagramu klas (nie korzystamy z większości dobrodziejstw tego diagramu, służy on jedynie prostej wizualizacji szkieletu naszej aplikacji). Wszystkie atrybuty są póki co publiczne ale we właściwej aplikacji ustawimy takie modyfikatory dostępu jakie będą potrzebne. Pominęliśmy również gettersy, settersy oraz pozostałe metody klas, ponieważ na tym etapie nie jesteśmy w stanie stwierdzić które z nich będą nam potrzebne, a które nie. Nie przedstawiliśmy na diagramie części klas, które znajdują się w naszej aplikacji (np. klasy związane z DAO, GUI). Ze względu na brak metod na diagramie ograniczyliśmy relacje między klasami do asocjacji i dziedziczenia.

Model został w dosyć naturalny sposób narzucony przez specyfikację projektu. Opis klas w nim zawartych wygląda następująco:

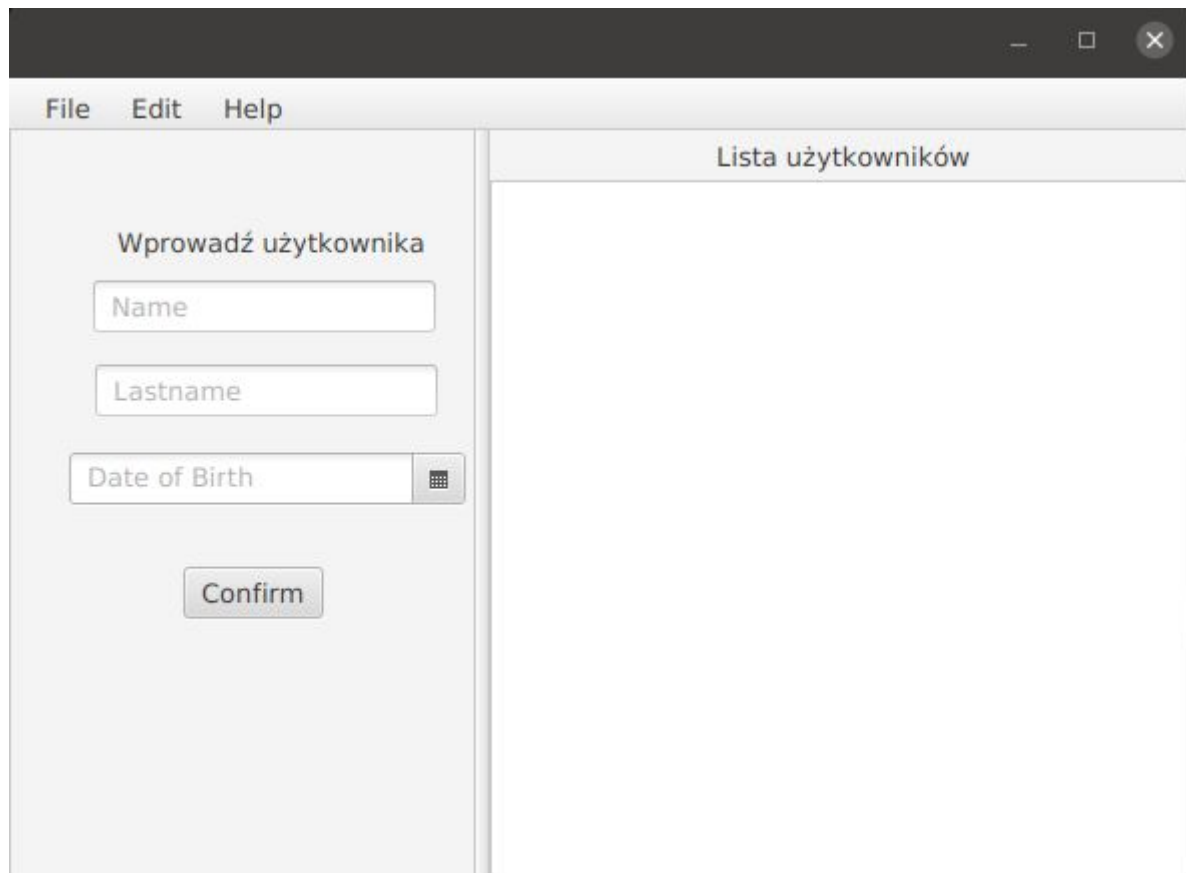
- Movie - Klasa reprezentująca film (rok powstania, tytuł oraz gatunek).
- MovieType - Enum reprezentujący typ filmu. Na chwilę obecną może to być film 2D lub 3D.
- Hall - Sala kinowa. Przechowuje informacje o limicie miejsc.
- Discount - Wszelkiego rodzaju promocje i upusty (wartość może być procentowa lub ilościowa).
- User - Użytkownik systemu.
- Customer - Użytkownik, który jest zwykłym klientem.
- Permissions - Enum związany z uprawnieniami pracowników
- Employee - Pracownik kina.
- Screening - Klasa reprezentująca konkretny seans w konkretnej sali i godzinie. Posiada np. informację i limicie miejsc na dany seans (może on być mniejszy bądź równy od limitu miejsca w sali), informację o cenie podstawowej biletu, która może być regulowana przez zniżki (klasa Discount).
- Ticket - Bilet kupiony (zarezerwowany) na jakieś miejsce na konkretnym seansie przez klienta

### **Baza danych i jej interfejs:**

Korzystamy z firestore - elastycznej nierelacyjnej bazy danej w chmurze (google). Klasa FirestoreDatabase jest używana do komunikacji z bazą i jest ona singletonem. Użyliśmy takiego rozwiązania, ponieważ klient bazy danych powinien być tylko jeden. Klasy HallDao oraz UserDao realizują wzorzec DAO, dzięki któremu możemy swobodnie manipulować bezpośrednią komunikacją z bazą nie zmieniając całej aplikacji.

### **GUI - dla osób:**

Wykonaliśmy GUI tylko dla dodawania użytkowników, ponieważ dla sal będzie to wyglądało analogicznie a nie mamy jeszcze planu jak zorganizować cały interfejs użytkownika



### Logika aplikacji:

Zdecydowaliśmy się na zorganizowanie struktury naszej aplikacji zgodnie ze wzorcem MVC, który jest wspierany przez JavaFX.

## M2

### 1. Planowany progres

- Działająca warstwa persystencji
- Autentykacja i autoryzacja
- Zarządzanie salami
- Zarządzenie filmami

### 2. Podział obowiązków

- Adam Bera - Kontrolery dla zarządzania salami i filmami
- Grzegorz Janosz - Napisanie pozostałych klas DAO, API do autoryzacji i autentykacji
- Krzysztof Hardek - Napisanie pozostałych klas modelu, widoki dla autentykacji i autoryzacji, zarządzania filmami, salami oraz dokumentacja

### 3. Co zostało zrobione

#### Model

Dopisane zostały pozostałe klasy z modelu.

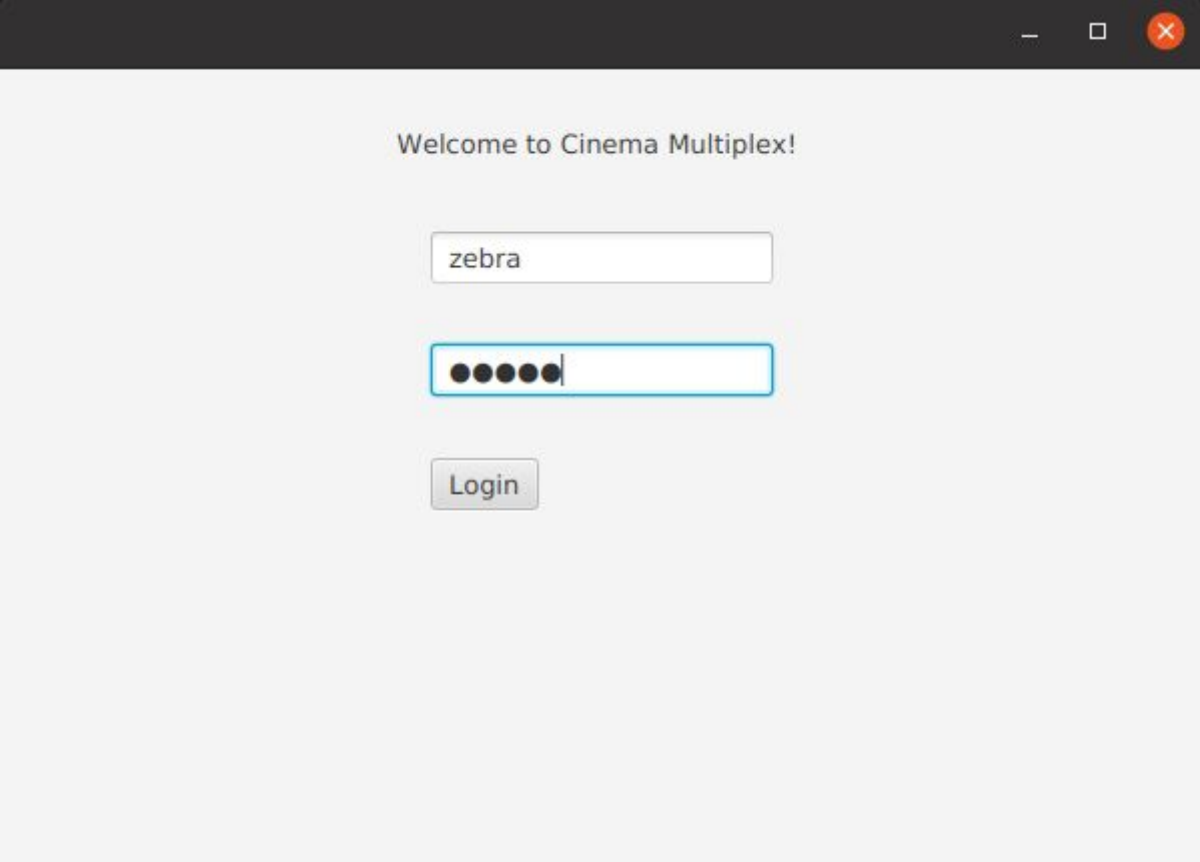
#### Interfejs danych i API do autoryzacji i autentykacji

Dopisane zostały pozostałe klasy DAO. Wystawiony został interfejs do autentykacji i autoryzacji. W wystawionym API do tworzenia queries wyników z bazy danych zostały użyte tzw. strategie szukające.

### **Widoki**

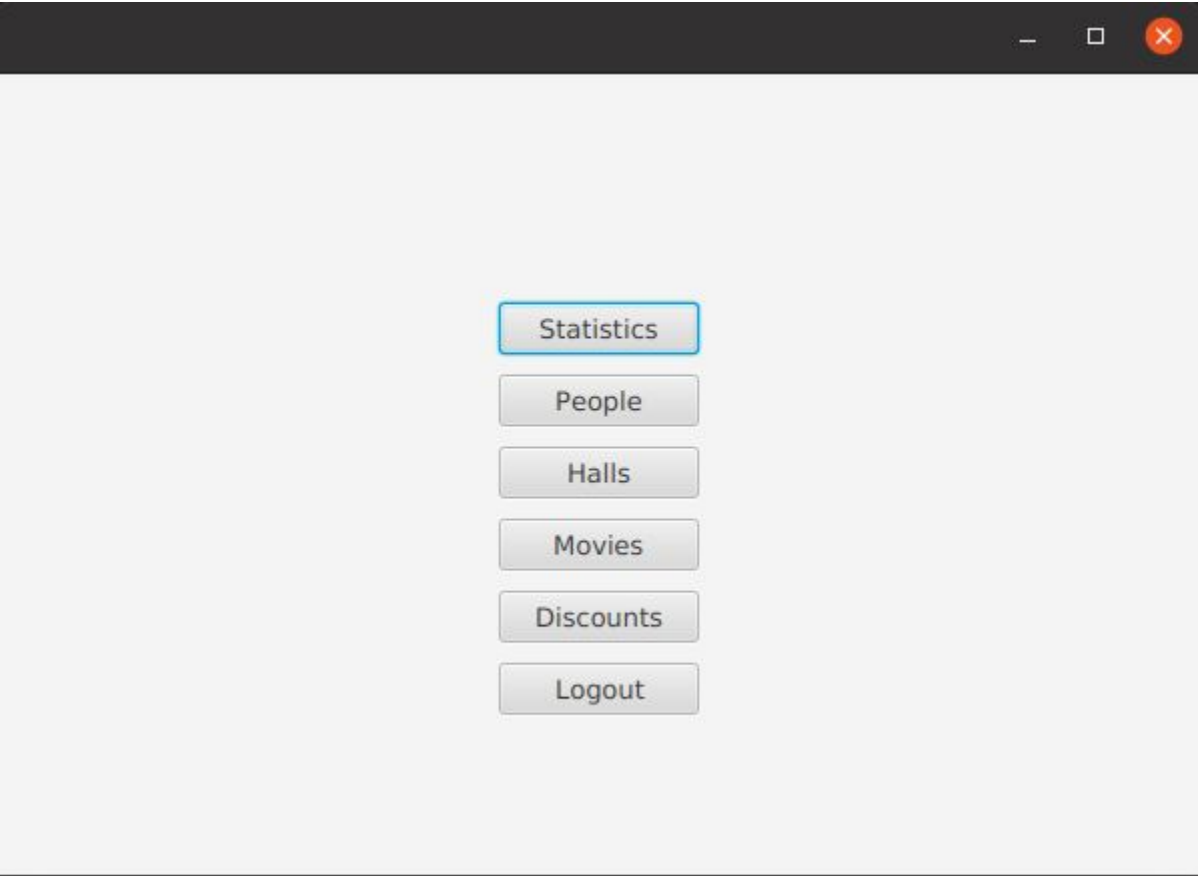
Dodane zostały widoki dla zarządzania salami i filmami (administrator), seansami (pracownik) oraz logowanie i panele główne dla administratora i pracownika

#### *- Login*

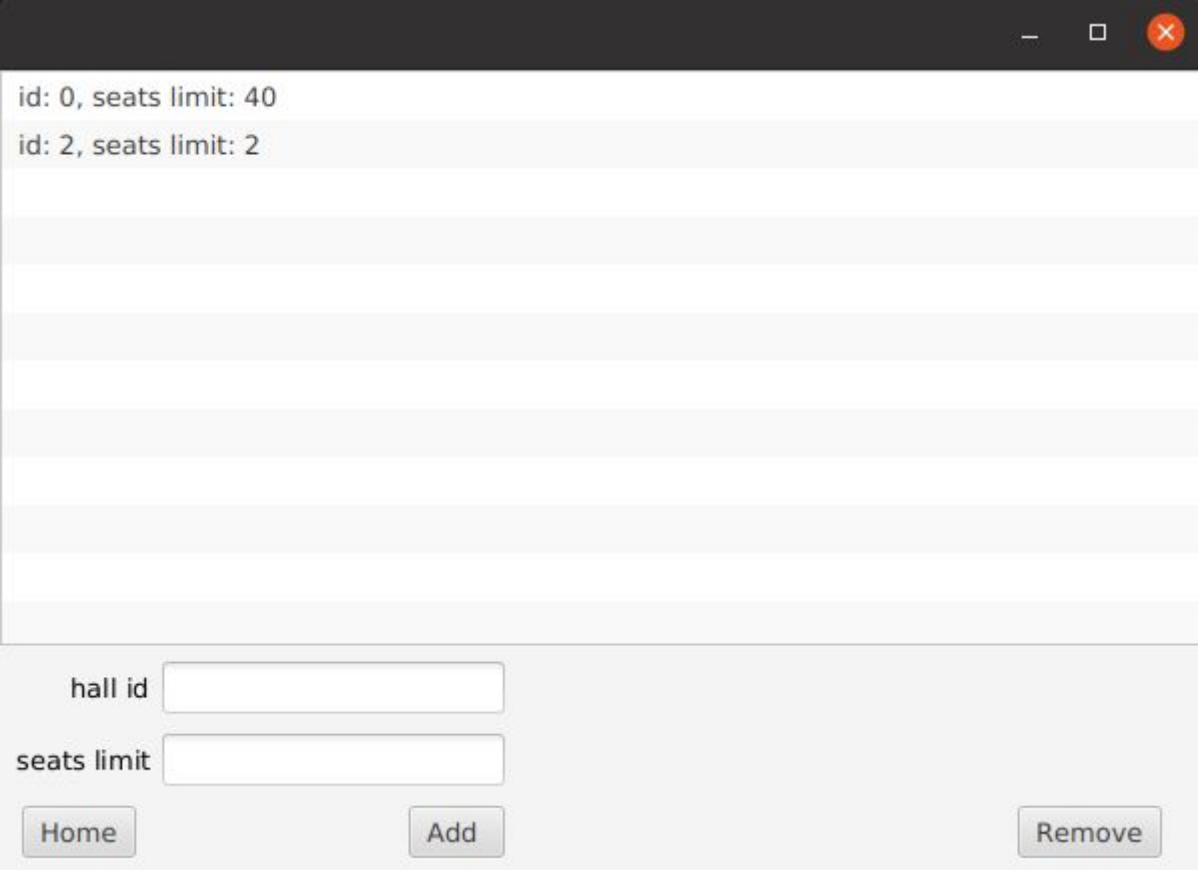


The screenshot shows a web application window titled "Welcome to Cinema Multiplex!". It features a login form with two input fields: a username field containing the text "zebra" and a password field with five black dots. Below the password field is a "Login" button. The window has a dark gray title bar with standard minimize, maximize, and close buttons.

#### *- Panel administratora*



- *Lista sal*



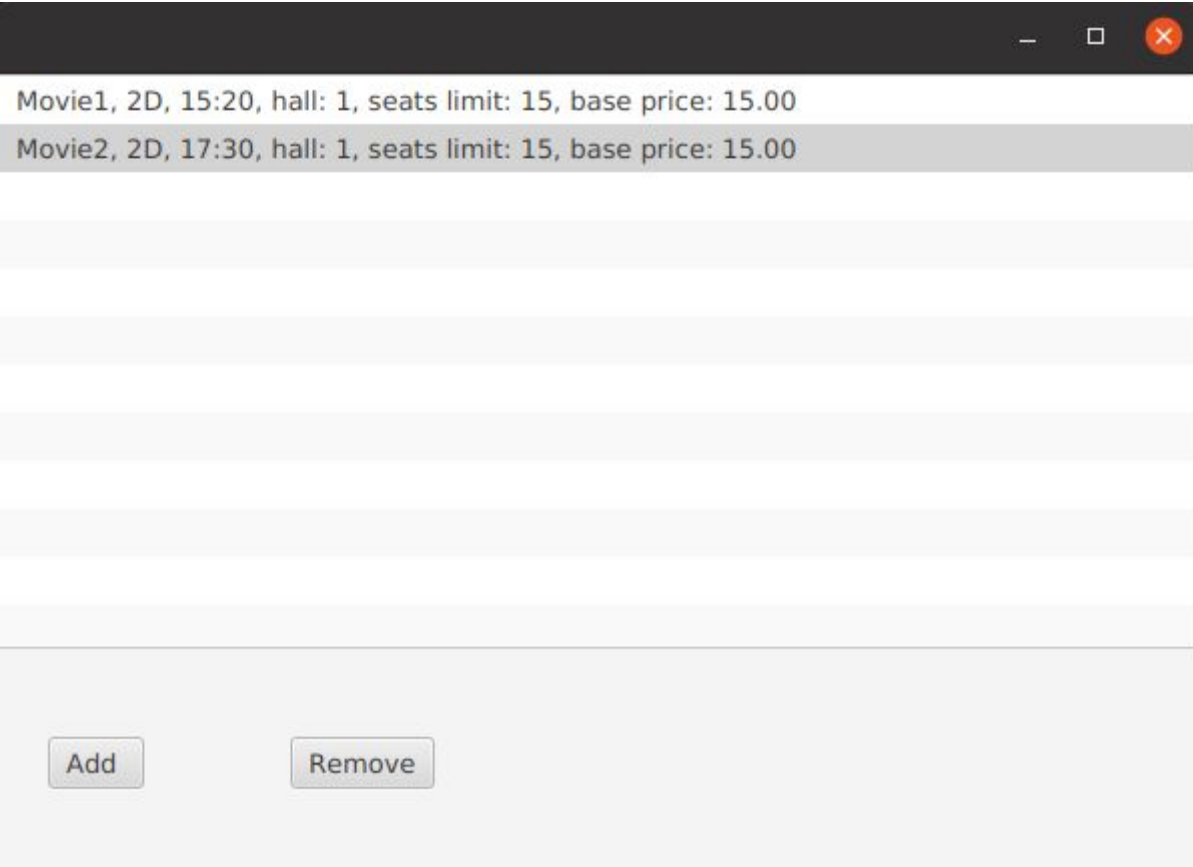
- *Lista filmów*

[illegible]

- *Panel pracownika*

A screenshot of a web application interface. At the top, there is a dark gray header bar containing three window control icons: a minus sign, a square, and a red circle with a white 'X'. The main content area is white and features three light gray buttons with rounded corners, arranged vertically in the center. The buttons are labeled 'Screenings', 'Tickets', and 'Logout' from top to bottom. The text on the buttons is in a dark gray, sans-serif font.

- *Lista seansów*



The screenshot shows a window titled "Lista seansów" with a dark header bar containing standard window controls (minimize, maximize, close). The main area is a list of movie showtimes, each on a separate row with alternating light and dark gray backgrounds. The first two rows contain data: "Movie1, 2D, 15:20, hall: 1, seats limit: 15, base price: 15.00" and "Movie2, 2D, 17:30, hall: 1, seats limit: 15, base price: 15.00". The remaining rows are empty. At the bottom of the window, there is a light gray bar containing two buttons: "Add" and "Remove".

Movie1, 2D, 15:20, hall: 1, seats limit: 15, base price: 15.00
Movie2, 2D, 17:30, hall: 1, seats limit: 15, base price: 15.00

Add Remove

- *Dodawanie seansów*



Movie

Date

Movie type

Time

Hall

Basic price

Add

- Dodawanie użytkowników

Login: adaber, Name: Adam, Surname: Bera

Login: addaac, Name: Ada, Surname: Bera

Login: grzjan, Name: Andrzej, Surname: Janosz

Login

Name

Surname

Add

Remove

Home

Logika

Zostały zrealizowane kontrolery dla widoków logowania, panelu administratora oraz zarządzania przez niego salami oraz filmami. Dodany został również kontroler do zarządzania użytkownikami, który nie został zrobiony wcześniej.

### M3

1. Planowany progres
2. Podział obowiązków
3. Co zostało zrobione