

Bazy Danych 2

Temat projektu: Kino

Natalia Górnikowska 259097

Mikołaj Pastuszek 259138

SPIS TREŚCI

1.	OPIS APLIKACJI.....	3
2.	DIAGRAM PRZYPADKÓW UŻYCIA.....	6
3.	DIAGRAM ZWIĄZEKÓW ENCIJI.....	14
4.	SFORMUŁOWANIE WYMAGAŃ DOTYCZĄCYCH DOSTĘPU DO BAZY I JEJ ZAWARTOŚCI.	16
5.	DIAGRAM KONCEPTUALNY.....	17
6.	OPRACOWANIE DIAGRAMU FIZYCZNEGO MODELU BAZY ORAZ UTWORZENIE SKRYPTU SQL DEFINIUJĄCEGO BAZĘ	18
7.	WDROŻENIE BAZY I PRZETESTOWANIE JEJ, W TYM SPRAWDZENIE:.....	24
8.	MAKIETA INTERFEJSU GRAFICZNEGO APLIKACJI BAZODANOWEJ NA PODSTAWIE SCENARIUSZY FUNKCJONOWANIA APLIKACJI BAZODANOWEJ I STRUKTURY BAZY.....	37
9.	DIAGRAMY CZYNNOŚCI DLA PRZYPADKÓW UŻYCIA: ALGORYTM FUNKCJONOWANIA APLIKACJI BAZODANOWEJ.....	46
10.	DIAGRAM KLAS I JEGO POWIĄZANIA W APLIKACJI BAZODANOWEJ	50
11.	TESTY.....	57
12.	DODATKI.....	81
13.	SPIS DIAGRAMÓW I RYSUNKÓW	82

1. Opis aplikacji

1.1. Opis zasobów ludzkich:

Aplikacja będzie przeznaczona do obsługi kina. Pracownicy placówki będą mogli dodawać do systemu nowo pożyczone przez kino filmy oraz tworzyć nowe seanse. Planowanie seansów będzie skupiało się na wybraniu filmu, zarezerwowaniu sali, która jest wolna w danym terminie, oraz ustaleniu ceny biletu wstępu. Bilety, które będą identyfikowane przez seans oraz wybrane miejsce w sali kinowej, będą mogły być sprzedawane fizycznie przez pracowników lub nabywane zdalnie przez klientów kina. Klient będzie miał możliwość zalogowania się do aplikacji, gdzie, oprócz zakupu biletu, będzie możliwa rezerwacja miejsc. W przypadku rezerwacji, gdy bilet nie zostanie zakupiony w kinie, rezerwacja wygasza godzinę przed seansem. Kierownik placówki będzie miał możliwość zatrudniania nowych pracowników oraz modyfikacji ich pensji, która jest zależna od zajmowanego stanowiska. Samo kino składa się kilku sali o różnej ilości miejsc.

1.2. Przepisy i strategia firmy:

Pracownik ponosi odpowiedzialność za poprawność danych, zalicza się do tego zgodność informacji dotyczących seansów z rzeczywistymi możliwościami kina. Pracownik odpowiada materialnie za wprowadzenie niepoprawnych danych, które mogą spowodować straty finansowe w jednostce gospodarczej.

1.3. Dane techniczne:

Klienci kina mogą za pośrednictwem aplikacji przeglądać, które miejsca na dany seans są wolne, a które zajęte. Zakłada się, że klientów równocześnie dokonujących tej operacji może być kilkudziesiąt. Kino składa się z jednej placówki ulokowanej we Wrocławiu.

1.4. Użytkownicy:

- Klienci
 - sprawdzają dostępność miejsc na dany seans
 - rezerwują miejsca na seanse
 - odstępują od rezerwacji
 - kupują bilety na seanse
 - logują się do systemu
- Pracownicy
 - sprzedają bilety na seanse
 - dodają nowo pożyczone filmy
 - tworzą nowe seanse
 - logują się do systemu
- Kierownik
 - dodaje i zwalnia pracowników
 - edytuje pensje pracowników
 - edytuje przyporządkowanych pracownikom stanowisk, które definiują ich kompetencje
 - loguje się do systemu
 - może wykonywać zadania przeznaczone dla pracowników

1.5. Lista wymagań funkcjonalnych

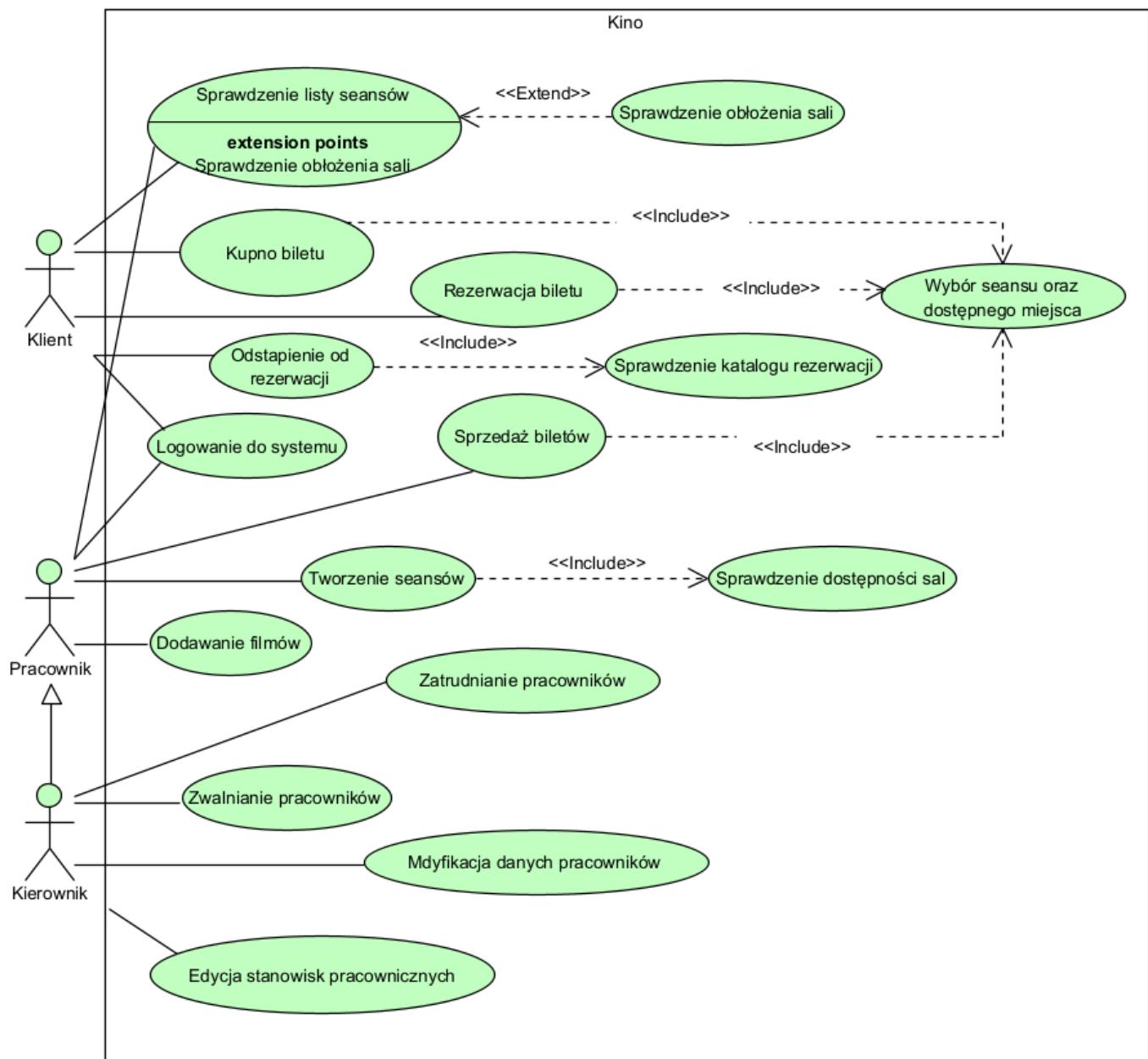
1. System zawiera katalog seansów, do którego można wstawić nowe seanse
2. Możliwe jest dokonywanie zakupu biletów w standardowej cenie lub z ulgą
3. System zawiera katalog zajętych miejsc na dany seans, na którego podstawie można ustalić dostępność miejsc przy zakupie biletów
4. Możliwe jest dokonywanie rezerwacji, która automatycznie zostaje usuwana godzinę przed seansem
5. Informacje o zajętych miejscach są automatycznie usuwane dzień dni po seansie, natomiast informacja o zakupach jest zachowywana w katalogu transakcji
6. System zawiera katalog pożyczanych przez kino filmów, do którego można dodawać informacje o tych zasobach

1.6. Lista wymagań niefunkcjonalnych

1. Dodawanie danych odnośnie pożyczanych przez kino filmów oraz tworzonych seansów może być dokonywane wyłącznie przez uprawnione do tego osoby

2. Modyfikacja listy pracowników, ich stanowisk czy pensji, może być dokonywana przez kierownika kina
3. Zakupy biletów może być dokonywany zdalnie przez klientów lub w kinie przez pracowników
4. Dokonywanie rezerwacji może być wykonywane jedynie przez klientów

2. Diagram przypadków użycia



Rysunek 1 - Diagram przypadków użycia

2.1. Definicje aktorów

AKTOR	OPIS	PRZYPADKI UŻYCIA
Klient	Klient, korzystając z programu, może dokonywać zakupów oraz rezerwacji biletów.	<ul style="list-style-type: none"> • PU Odstępnie od rezerwacji powiązane przez <<include>> z PU Sprawdzenie katalogu rezerwacji • PU Sprawdzenie listy seansów powiązane przez <<extend>> z PU Sprawdzenie obłożenia sali • PU Rezerwacja biletu powiązane przez <<include>> z PU Wybór seansu oraz dostępnego miejsca • PU Kupno biletu powiązane przez <<include>> z PU Wybór seansu oraz dostępnego miejsca • PU Logowanie do systemu
Pracownik	Pracownik dodaje nowe pożyczone filmy, tworzą nowe seanse oraz sprzedają bilety.	<ul style="list-style-type: none"> • PU Logowanie do systemu • PU Sprzedaż biletów powiązane przez <<include>> z PU Wybór seansu oraz dostępnego miejsca • PU Tworzenie seansów powiązane przez <<include>> z PU Sprawdzenie dostępności sali • PU Dodawanie filmów • PU sprawdzanie listy seansów powiązane przez <<extend>> z PU Sprawdzenie obłożenia sali
Kierownik	Kierownik może dodatkowo edytować listę i dane pracowników oraz dane donośnie stanowisk pracowniczych.	<ul style="list-style-type: none"> • PU Zatrudnianie pracowników • PU Zwalnianie pracowników • PU Modyfikacja danych pracowników • PU Edycja stanowisk

2.2. Opisy przypadków użycia

a) PU Sprawdzenie listy seansów

OPIS

Cel: Sprawdzenie jakie seanse są zaplanowane przez kino

Punkty rozszerzenia: PU Sprawdzenie obłożenia sali

Aktorzy: Klient, Pracownik, Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: brak

Warunki końcowe: wyświetlenie listy seansów lub pustej listy seansów

Przebieg:

1. Wyszukanie i wyświetlenia listy seansów na najbliższe dni
2. Możliwość wyboru seansu oraz sprawdzenia obłożenia sali (*PU Sprawdzenie obłożenia sali*)

b) **PU Sprawdzenie obłożenia sali**

OPIS

Cel: Sprawdzenie dostępności miejsc na dany seans

Aktorzy: Klient, Pracownik, Kierownik

Zdarzenie inicjujące: wywoływany przez PU Sprawdzenie listy seansów

Warunki wstępne: istnienie danego seansu

Warunki końcowe: wyświetlenie listy dostępnych miejsc

Przebieg:

1. Wybór seansu zgodnie z listą seansów
2. Wyświetlenie wolnych miejsc w sali

c) **PU Kupno biletu**

OPIS

Cel: Zakup biletu na dany seans

Aktorzy: Klient

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: istnienie seansów, klienta jest zalogowany

Warunki końcowe: wyświetlenie informacji o niepowodzeniu operacji lub o zakupie biletu

Przebieg:

1. Wywołanie PU Wybór seansu oraz dostępnego miejsca.
2. Wybór biletu z ulgą lub bez
3. Zatwierdzenie transakcji kupna biletu
4. Wyświetlenie komunikatu o pomyślnej transakcji

d) **PU Wybór seansu oraz dostępnego miejsca**

OPIS

Cel: Wybór seansu z listy oraz miejsca na danym seansie

Aktorzy: Klient, Pracownik, Kierownik

Zdarzenie inicjujące: wywoływanego przez PU Kupno biletu, PU Rezerwacja biletu lub PU Sprzedaż biletów

Warunki wstępne: istnienie seansów

Warunki końcowe: wybór miejsca lub wyświetlenie informacji o braku wolnych miejsc

Przebieg:

1. Wybór seansu z listy
2. Wyszukanie wolnych miejsc
3. Wyświetlenie listy dostępnych miejsc
4. Wybranie miejsca z listy

e) **PU Rezerwacja biletu**

OPIS

Cel: Rezerwacja biletu na seans

Aktorzy: Klient

Zdarzenie inicjujące: inicjalizacja przez uruchomienie systemu

Warunki wstępne: aktor musi być zalogowany, istnienie seansów

Warunki końcowe: wyświetlenie informacji o niepowodzeniu operacji lub o dokonaniu rezerwacji

Przebieg:

1. Wywołanie PU Wybór seansu oraz dostępnego miejsca
 - a. W przypadku braku wolnych miejsc należy zakończyć PU
 - b. W przypadku wyboru miejsc, należy zatwierdzić chęci dokonania rezerwacji

f) **PU Odstąpienie od rezerwacji**

OPIS

Cel: Odstąpienie od wcześniejszej rezerwacji

Aktorzy: Klient

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

Warunki końcowe: usunięcie rezerwacji miejsca, jeśli była taka rezerwacja

Przebieg:

1. Należy wywołać *PU Sprawdzenie katalogu rezerwacji*, w celu wyszukania istniejących rezerwacji.
2. Wyświetlenie listy rezerwacji klienta
3. Wybranie rezerwacji z listy
4. Jeśli rezerwacja została wybrana, a operacja zatwierdzona należy usunąć rezerwację.

g) **PU Sprawdzenia katalogu rezerwacji**

OPIS

Cel: Sprawdzenie zarezerwowanych miejsc przez danego klienta

Aktorzy: Klient

Zdarzenie inicjujące: jest wywołany z *PU Sprawdzenie katalogu rezerwacji*

Warunki wstępne: brak

Warunki końcowe: wyszukanie istniejących rezerwacji klienta

Przebieg:

1. Szukanie rezerwacji według danych podanych do przypadku użycia
2. Jeśli istnieją rezerwacje o podanych danych, jest ona zwracana, w przeciwnym wypadku zwracana jest informacja o braku rezerwacji

h) **PU Logowanie do systemu**

OPIS

Cel: zalogowanie się do systemu

Aktorzy: Klient, Pracownik, Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: brak

Warunki końcowe: zalogowanie aktora lub komunikat o niepowodzeniu

Przebieg:

1. Podanie loginu oraz hasła
2. Sprawdzenie poprawności danych.
 - a. Rozpoczęcie logowania do systemu w przypadku poprawnych danych
 - b. Pojawienie się komunikatu o podaniu niepoprawnych danych

i) **PU Sprzedaż biletów**

OPIS

Cel: Sprzedaż biletów

Aktorzy: Pracownik, Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

Warunki końcowe: wyświetlenie komunikatu o powodzeniu lub niepowodzeniu operacji

Przebieg:

1. Wywołanie PU Wybór seansu oraz dostępnego miejsca
2. Wybór biletu z ulgą lub bez

j) **PU Tworzenie seansów**

OPIS

Cel: Dodawanie nowych seansów

Aktorzy: Pracownik, Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

Warunki końcowe: dodanie seansu do katalogu seansów lub wyświetlenie komunikatu o niepowodzeniu operacji

Przebieg:

1. Wybór filmu z listy filmów, wybranie sali z listy z salami podanie daty rozpoczęcia i zakończenia seansu oraz ceny biletu wstępu.
2. Zatwierdzenie podanych.
3. Wywołanie PU Sprawdzenie dostępności sali, w celu sprawdzenie dostępności wybranej sali, a na tej podstawie również terminu seansu.
4. W przypadku poprawności danych dodanie seansu, w przeciwnym przypadku wyświetlenie komunikatu o niepowodzeniu operacji

k) **PU Sprawdzenie dostępności sali**

OPIS

Cel: Sprawdzenie terminów dostępności sali

Aktorzy: Pracownik, Kierownik

Zdarzenie inicjujące: wywoływanego przez PU Tworzenie seansów

Warunki wstępne: brak

Warunki końcowe: zwrócenie informacji o wybranej dostępności sali

Przebieg:

1. Sprawdzenie dostępności sali w wybranym terminie
2. Rezerwacja sali w wybranym terminie

I) PU Dodawanie filmów

OPIS

Cel: dodawanie nowo pożyczonych przez kino filmów

Aktorzy: Pracownik, Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

Warunki końcowe: dodanie filmu do listy

Przebieg:

1. Podanie atrybutów charakteryzujących film: tytuł, czas trwania, dystrybutora, datę pożyczania, datę oddania oraz informacji technicznych czy film jest 3D oraz czy posiada lektora bądź napisy
2. Zatwierdzenie operacji

m) PU Zatrudnianie pracowników

OPIS

Cel: zatrudnianie nowych pracowników

Aktorzy: Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

Warunki końcowe: dodanie pracownika do listy pracowników lub wyświetlenie się komunikatu o niepowodzeniu operacji

Przebieg:

1. Należy podać dane pracownika takie jak: imię, nazwisko, nr telefonu, datę zatrudnienia oraz login i hasło
2. Dodanie pracownika lub w przypadku, gdy pracownik istnieje pojawiienie się komunikatu o niepowodzeniu operacji

n) PU Zwalnianie pracowników

OPIS

Cel: zwalnianie pracowników

Aktorzy: Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: pracownik istnieje w systemie, aktor musi być zalogowany

Warunki końcowe: dodanie daty zwolnienia dla danego pracownika

Przebieg:

1. Wybranie pracownika do zwolnienia.
2. Zatwierdzenie operacji i dodanie daty zwolnienia do danych pracownika

o) PU Modyfikacja danych pracowników

OPIS

Cel: Aktualizacja zapisanych danych odnośnie pracowników

Aktorzy: Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: widnienie pracownika w systemie

Warunki końcowe: zapisanie zmienionych danych

Przebieg:

1. Wybranie danych do zmiany
2. Następnie należy podać poprawnie dane i zaakceptować

p) PU Edycja stanowisk pracowników

OPIS

Cel: Edycja danych dotyczących stanowisk (np. pensji przypisanej do stanowiska)

Aktorzy: Kierownik

Zdarzenie inicjujące: inicjalizacja przez uruchomienie programu

Warunki wstępne: aktor musi być zalogowany

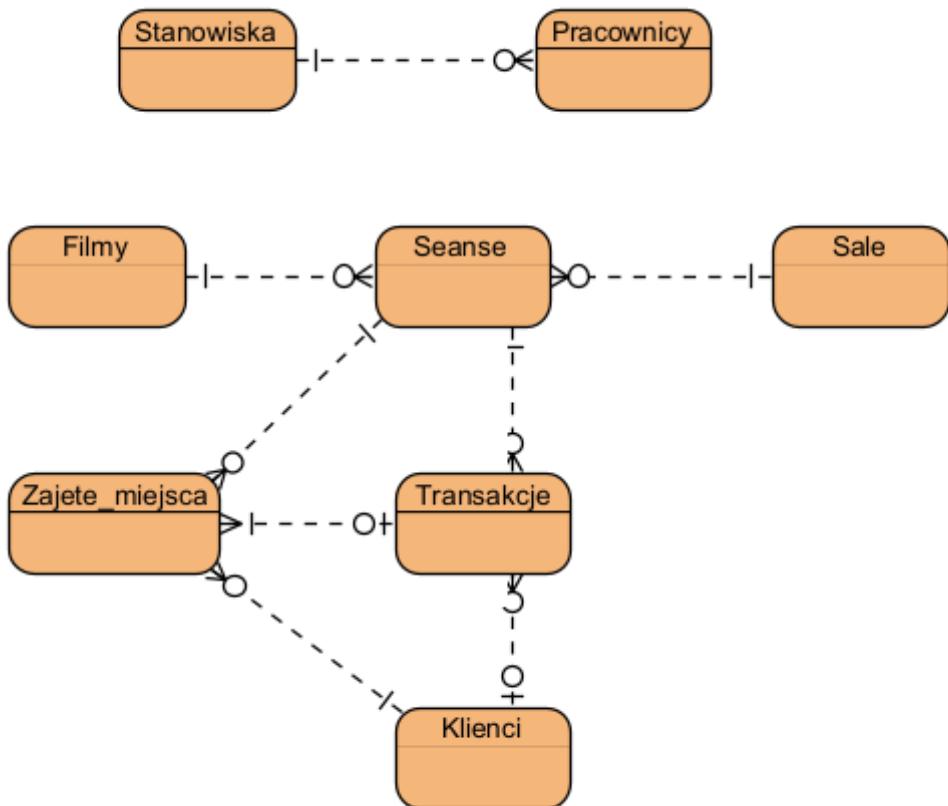
Warunki końcowe: zapisanie zmian

Przebieg:

1. Należy podać nazwę stanowiska, które będzie edytowane
 - a. Jeżeli stanowisko nie istnieje, może zostać dodane.
 - b. Jeżeli stanowisko istnieje, należy wybrać kategorię do edycji oraz podać nową wartość.

3. Diagram związków encji

3.1. Poglądowy ERD



Rysunek 2 - Poglądowy ERD

W tabeli **Zajete_miejsca** znajdują się rekordy miejsc zajętych bądź zarezerwowanych. Rezerwacje usuwają się samoistnie godzinę przed seansem natomiast miejsca zajęte są usuwanie dzień po seansie. Tabela przechowuje dane tymczasowe i powstała, by ułatwić przeszukiwanie na bieżąco bazy danych w sprawie informacji o aktualnie dostępnych biletach. Natomiast dane odnośnie dokonanych przez klientów zakupów są przechowywane trwale w tabeli **Transakcje**.

3.2. Tabele:

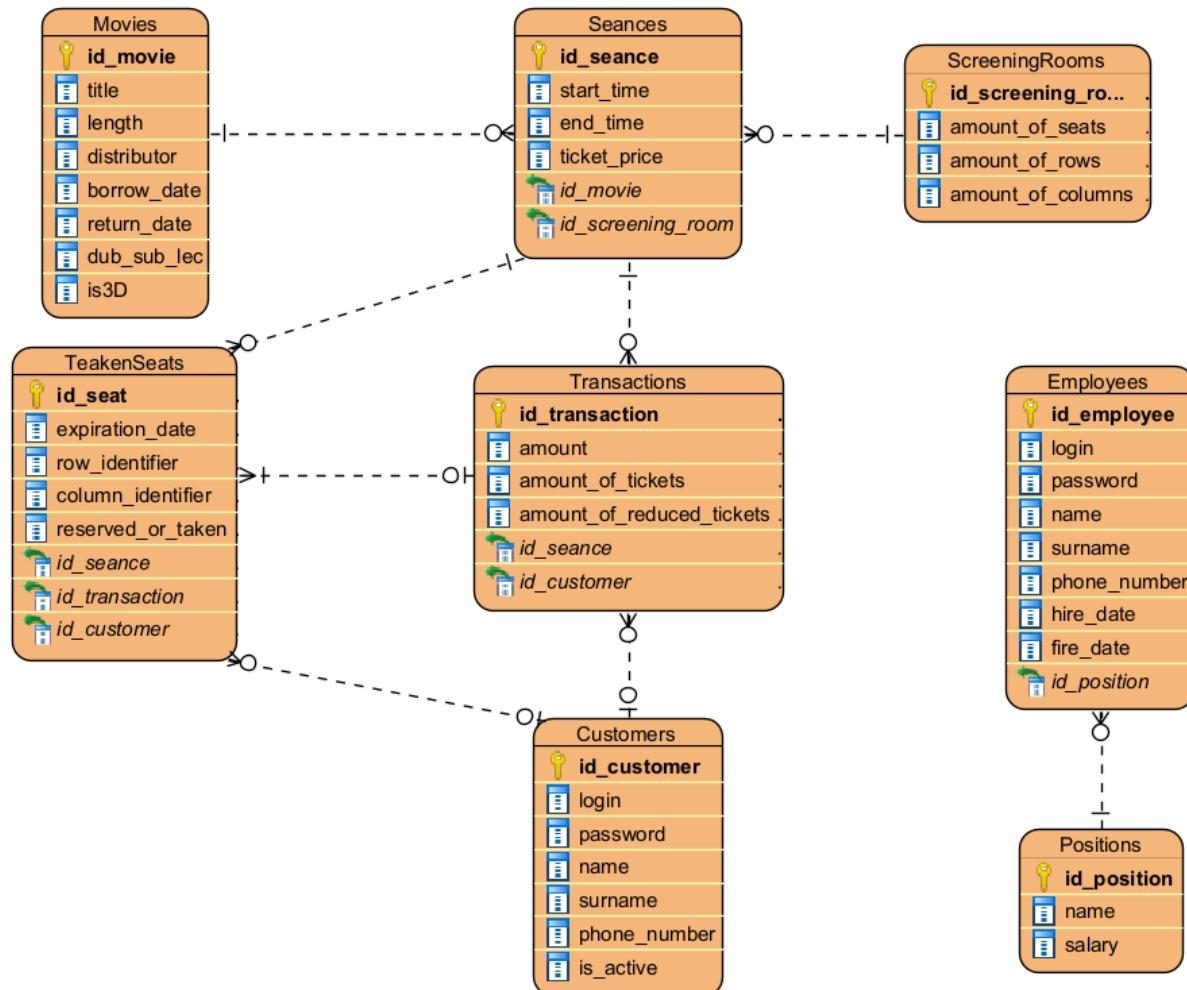
- Filmy: **id_filmu**, tytul, czas trwania, dystrybutor, data pozyczenia, data oddania, napisy/lektor, czy 3D
- Sale: **id_sali**, liczba miejsc, liczba rzędow, liczba miejsc w rzędzie
- Stanowiska: **id_stanowiska**, nazwa, pensja
- Pracownicy: **id_pracownika**, login, hasło, imię, nazwisko, id_stanowiska, nr_telefonu, data_zatrudnienia, data_zwolnienia
- Klienci: **id_klienta**, login, haslo, imie, nazwisko, nr_telefonu

- Seanse: **id_seansu**, id_filmu, id_sali, rozpoczęcie, zakończenie, cena
- Zajęte_miejsca: **id_miejsca**, id_seansu, id_transakcji, data ważności, rząd, miejsce w rzędzie, rezerwacja/ zajęte, id_klienta
- Transakcje: **id_transakcji**, id_seansu, id_klienta, kwota, liczba_biletow, liczba_ulgowych

4. Sformułowanie wymagań dotyczących dostępu do bazy i jej zawartości.

Tabela	Wstawianie	Modyfikacja	Usuwanie	Wyszukiwanie
Movies	Pracownik, kierownik	Pracownik, kierownik	-	Pracownik, kierownik
ScreeningRooms	Kierownik	-	-	Pracownik, kierownik
Positions	Kierownik	Kierownik	-	Kierownik
Employees	Kierownik	Kierownik	-	Kierownik
Customers	Klient	Klient	-	Pracownik, Kierownik
Seances	Pracownik, kierownik	Pracownik, kierownik	-	Pracownik, kierownik, klient
TeakenSeats	Pracownik, kierownik, klient	Pracownik, kierownik, klient	Klient	Pracownik, kierownik, klient
Transactions	Pracownik, kierownik, klient	-	-	Pracownik, kierownik

5. Diagram konceptualny



Rysunek 3 - Konceptualny ERD

6. Opracowanie diagramu fizycznego modelu bazy oraz utworzenie skryptu SQL definiującego bazę

6.1. Zdefiniowanie użytkowników bazy danych i nadanie im odpowiednich uprawnień.

Dla bazy danych zostali zidentyfikowani dwaj użytkownicy. Pierwszym jest „kino”, jest to konto, w którym jest wywoływany cały skrypt bazy. Wspomnianym kontem zarządza administrator. Drugim użytkownikiem jest „cinema_user”. Jest to konto połączone z aplikacją.

Tworzenie użytkownika „kino” przez system:

```
create user kino identified by kino;
GRANT CONNECT TO kino WITH ADMIN OPTION;
GRANT ALL PRIVILEGES TO kino;
```

Tworzenie użytkownika „cinema_user” przez użytkownika „kino”:

```
-----TWORZENIE UZYTKOWNIKA-----
create user c##cinema_user identified by cinema_user;
GRANT CONNECT TO c##cinema_user;

---uprawnienia do procedur---
GRANT execute on changeCustomer to c##cinema_user;
GRANT execute on changePositionName to c##cinema_user;
GRANT execute on changePositionSalary to c##cinema_user;
GRANT execute on changeReservationToTeaken to c##cinema_user;
GRANT execute on customerAccountDeactivation to c##cinema_user;
GRANT execute on deleteReservation to c##cinema_user;
GRANT execute on fireEmployee to c##cinema_user;
GRANT execute on insertCustomer to c##cinema_user;
GRANT execute on insertEmployee to c##cinema_user;
GRANT execute on insertMovie to c##cinema_user;
GRANT execute on insertPosition to c##cinema_user;
GRANT execute on insertResTeakenSeats to c##cinema_user;
GRANT execute on insertScreeningRoom to c##cinema_user;
GRANT execute on insertSeance to c##cinema_user;
GRANT execute on insertTeakenSeats to c##cinema_user;
GRANT execute on insertTransactions to c##cinema_user;
GRANT execute on logInCustomer to c##cinema_user;
GRANT execute on logInEmployee to c##cinema_user;
GRANT execute on isEmployeeAnManager to c##cinema_user;
GRANT execute on getTransactionId to c##cinema_user;

---uprawnienia do widokow---
GRANT select on ViewCustomers to c##cinema_user;
GRANT select on ViewEmployees to c##cinema_user;
GRANT select on ViewMovies to c##cinema_user;
GRANT select on ViewPositions to c##cinema_user;
GRANT select on ViewScreeningrooms to c##cinema_user;
GRANT select on ViewSeances to c##cinema_user;
GRANT select on ViewTeakenseats to c##cinema_user;
GRANT select on ViewTransactions to c##cinema_user;
```

Użytkownik dostał dostęp do wglądu do tabel oraz uprawnienia do wykonywania procedur, które mogą modyfikować zawartości tabel.

6.2. Zdefiniowanie typów kolumn zgodnie z wymaganiami serwera.

Tabele w bazie zostały stworzone na podstawie diagramu ERD przedstawione na stronie numer 8.

Przykładowe zdefiniowanie tabeli w skrypcie:

```
CREATE TABLE Movies (
    id_movie number(10),
    title varchar2(50 BYTE) NOT NULL,
    length number(3) NOT NULL,
    distributor varchar2(50 BYTE) NOT NULL,
    borrow_date date NOT NULL,
    return_date date,
    dub_sub_lec char(1) NOT NULL
    CHECK (dub_sub_lec = 'D' OR dub_sub_lec = 'S' OR dub_sub_lec = 'L'),
    is3D number(1) NOT NULL CHECK(is3D = 0 OR is3D = 1),
    PRIMARY KEY(id_movie)
);
```

Link do reszty skryptu będzie udostępniony w punkcie „5. Dodatki”.

6.3. Ustawienie odpowiednich kolumn jako unikalne i/lub opcjonalne.

Dla każdej zdefiniowanej tabeli zostało ustalone, które z kolumn są unikalne lub opcjonalne, co można zobaczyć na diagramie fizycznym zaprezentowany na stronie 8.

Przykładowe zdefiniowanie kolumny jako unikalnej oraz obligatoryjnej (dla tabeli Employees):

```
login varchar2(50 BYTE) UNIQUE NOT NULL,
```

6.4. Zdefiniowanie kluczy głównych i obcych dla powiązania tabel.

Wszystko można zobaczyć na diagramie fizycznym zaprezentowany na stronie 8.

Przykładowe zdefiniowanie klucza głównego tabeli Seances:

```
PRIMARY KEY(id_seance),
```

Przykładowe zdefiniowania kluczy obcych tabeli Seances:

```

CONSTRAINT FK_Movies_Seances
    FOREIGN KEY (id_movie)
        REFERENCES Movies(id_movie),
CONSTRAINT FK_ScreeningRooms_Seances
    FOREIGN KEY (id_screening_room)
        REFERENCES ScreeningRooms(id_screening_room)

```

6.5. Zdefiniowanie zachowania bazy w przypadku modyfikacji i usuwania danych w powiązanych tabelach, z wykorzystaniem wyzwalaczy.

W bazie danych zostało stworzone 8 wyzwalaczy, każdy jest przypisany do jednej tabeli. Zostały one stworzone, by uruchamiać sekwencje, które auto inkrementują klucz główny w poszczególnych tabelach. Poniżej zaprezentowane zostało tworzenie jednego z nich, dokładnie dla tabeli Movies i klucza id_movies.

```

CREATE SEQUENCE MOVIES_ID_SEQ
    START WITH 1 INCREMENT BY 1 CACHE 20;

create or replace TRIGGER MOVIES_ID_TRG
BEFORE INSERT ON MOVIES
FOR EACH ROW
BEGIN
<<COLUMN_SEQUENCES>>
BEGIN
    IF INSERTING AND :NEW.ID_MOVIE IS NULL THEN
        SELECT MOVIES_ID_SEQ.NEXTVAL INTO :NEW.ID_MOVIE FROM SYS.DUAL;
    END IF;
END COLUMN_SEQUENCES;
END;
/

```

Jednoczesnym dodawaniem danych do tabel, jak jest to w sytuacji dodawania rekordu do tabeli transakcji po zakupie biletów zapisywanych do tabeli zajętych miejsc, zajmują się procedury bądź w przyszłości warstwa aplikacji zapisana w języku JAVA. Zgodnie z wcześniej zaprezentowanymi warunkami oraz dokładną analizą, zostało ustalone, iż usuwanie rekordów z tabeli, jest możliwe wyłącznie w przypadku tabeli TakenSeats. Jest to przeprowadzane wyłącznie za pomocą procedury, ponieważ nie wymaga to zmian w innych tabelach.

6.6. Zdefiniowanie procedur lub funkcji dostępu do danych (automatyzacja i bezpieczeństwo wykonywania zapytań na bazie).

Nazwa procedury	Co robi dana procedura
insertResTeakenSeats	Dodaje dane do tabeli TeakenSeats. Ustawia wartości expiration_date na godzinę przed startem seansu oraz reserved_or_teaken na R.
insertTeakenSeats	Dodaje dane do tabeli TeakenSeats. Ustawia wartości expiration_date na 2 godzinę po zakończeniu seansu oraz reserved_or_teaken na T.
insertTransactions	Dodaje dane do tabeli Transactions.
insertMovie	Dodaje dane do tabeli Movie
insertSeance	Dodaje dane do tabeli Seance. W razie podania końca seansu krótszego niż minimalna wartość (początek sensu + długość filmu), procedura automatycznie zmieni koniec seansu na minimalną wartość.
insertScreeningRoom	Dodaje dane do tabeli ScreeningRoom
insertCustomer	Dodaje dane do tabeli Customer
insertEmployee	Dodaje dane do tabeli Employee
insertPosition	Dodaje dane do tabeli Position
changeCustomer	Zmienia wartości w tabeli Customer w kolumnach password, name, surname oraz phone_number, dla pojedynczego klienta.
customerAccountDeactivation	Zmienia wartości w tabeli Customer w kolumnie is_active na wartość 0, dla pojedynczego klienta.
fireEmployee	Modyfikuje wartość w tabeli Employees w kolumnie fire_date, dla pojedynczego pracownika.
changePositionName	Zmienia wartość w tabeli Positions w kolumnie name, dla pojedynczej pozycji.
changePositionSalary	Zmienia wartość w tabeli Positions w kolumnie salary, dla pojedynczej pozycji.
changeReservationToTeaken	Zmienia wartość na 'T' w tabeli TeakenSeats w kolumnie reserved_or_teaken, dla pojedynczego miejsca.
deleteReservation	Usuwa rekord w tabeli TeakenSeats, który oznaczał zarezerwowane miejsce.
deleteTS	Usuwa wszystkie rekordy z tabeli TeakenSeats, gdzie wartość kolumny expiration_date jest mniejsza od aktualnej daty. Procedura jest wywoływana przez joba co 5 min.

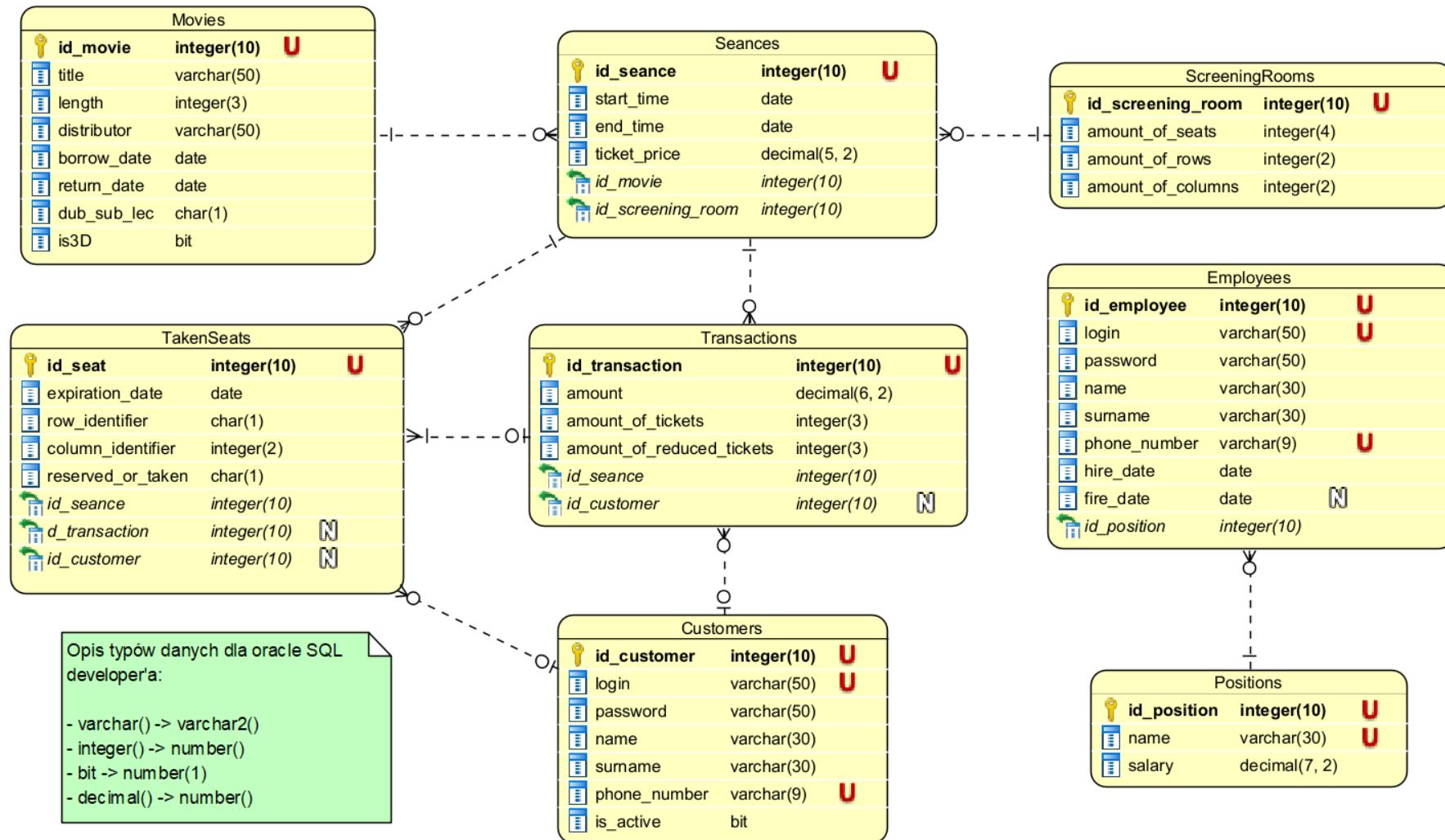
Testy oraz zdjęcia implementacji wszystkich procedur znajdują się w punkcie 4 d).

6.7. Wykorzystanie innych możliwości wybranego systemu zarządzania bazą.

W bazie danych został dodany job o nazwie clearTeakenSeats. Służy on do cyklicznego wywoływania procedury deleteTS, która usuwa niepotrzebne wiersze z tabeli TeakenSeats. Job uruchamia procedurę co 5 min.

```
BEGIN
    DBMS_SCHEDULER.create_job (
        job_name      => 'clearTeakenSeats',
        job_type      => 'PLSQL_BLOCK',
        job_action    => 'BEGIN deleteTS; END',
        start_date    => SYSTIMESTAMP,
        repeat_interval => 'FREQ=MINUTELY; INTERVAL=5',
        enabled       => TRUE
    );
END;
/
```

6.8. Diagram fizyczny:



Rysunek 4 - Fizyczny ERD

7. Wdrożenie bazy i przetestowanie jej, w tym sprawdzenie:

7.1. integralności semantycznej (czy dane mają właściwy typ, rozmiar, format itp.);

Przykładowe testowanie integralności semantycznej zostało przeprowadzone dla tabeli Movies.

Poniżej zaprezentowane są typy danych wszystkich kolumn tabeli oraz nałożone dla nich ograniczenia.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID
1	ID_MOVIE	NUMBER(10,0)	No	(null)	1
2	TITLE	VARCHAR2(50 BYTE)	No	(null)	2
3	LENGTH	NUMBER(3,0)	No	(null)	3
4	DISTRIBUTOR	VARCHAR2(50 BYTE)	No	(null)	4
5	BORROW_DATE	DATE	No	(null)	5
6	RETURN_DATE	DATE	Yes	(null)	6
7	DUB_SUB_LEC	CHAR(1 BYTE)	No	(null)	7
8	IS3D	NUMBER(1,0)	No	(null)	8

```
1 "TITLE" IS NOT NULL
2 "LENGTH" IS NOT NULL
3 "DISTRIBUTOR" IS NOT NULL
4 "BORROW_DATE" IS NOT NULL
5 "DUB_SUB_LEC" IS NOT NULL
6 "IS3D" IS NOT NULL
7 dub_sub_lec = 'D' OR dub_sub_lec = 'S' OR dub_sub_lec = 'L'
8 is3D = 0 OR is3D = 1
```

Przy pomocy poniższej komendy została dokonana próba dodania danych do tabeli, która zakończyła się sukcesem. W konsekwencji w konsoli pojawił się komunikat o poprawnym dodaniu rekordu.

Pokazało to poprawność doboru typów danych dla konkretnych kolumn.

```
INSERT INTO Movies (title, length, distributor, borrow_date, return_date,
dub_sub_lec, is3D)
VALUES ('Superman',140, 'xx distributor', to_date('2022-01-02', 'YYYY-MM-DD'),
to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);
```

```
1 row inserted.
```

7.2. integralności encji (czy tabele mają unikalne i niepuste klucze główne);

Jako przykład, do zbadania czy integralność encji została zachowana również posłuży tabela Movies.

W przypadku dodania dwóch rekordów bez podania wartości klucza głównego, nastąpiło automatyczne dodanie auto-inkrementującego się klucza, co umożliwiły dodane do projektu wyzwalacze oraz sekwencje.

Komenda:

```
INSERT INTO Movies (title, length, distributor, borrow_date, return_date,
    dub_sub_lec, is3D)
VALUES ('Superman',140, 'xx distributor', to_date('2022-01-02', 'YYYY-MM-DD'),
    to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);

INSERT INTO Movies (title, length, distributor, borrow_date, return_date,
    dub_sub_lec, is3D)
VALUES ('Spiderman',120, 'xx distributor', to_date('2022-01-04', 'YYYY-MM-DD'),
    to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);
```

Wynik działania:

ID_MOVIE	TITLE	LENGTH	DISTRIBUTOR	BORROW_DATE	RETURN_DATE	DUB_SUB_LEC	IS3D
1	1 Superman	140	xx distributor	22/01/02	22/02/04	D	0
2	2 Spiderman	120	xx distributor	22/01/04	22/02/04	D	0

W przypadku ręcznego podania wartości klucza głównego, który dla obu rekordów przyjął tą samą wartość, został wyświetlony błąd, a dane nie zostały dodane do tabeli.

Komenda:

```
INSERT INTO Movies (id_movie, title, length, distributor, borrow_date, return_date,
    dub_sub_lec, is3D)
VALUES (1,'Superman',140, 'xx distributor', to_date('2022-01-02', 'YYYY-MM-DD'),
    to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);

INSERT INTO Movies (id_movie, title, length, distributor, borrow_date, return_date,
    dub_sub_lec, is3D)
VALUES (1,'Spiderman',120, 'xx distributor', to_date('2022-01-04', 'YYYY-MM-DD'),
    to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);
```

Komunikat:

```
Error starting at line : 626 in command -
INSERT INTO Movies (id_movie, title, length, distributor, borrow_date, return_date,
       dub_sub_lec, is3D)
VALUES (1,'Spiderman',120, 'xx distributor', to_date('2022-01-04', 'YYYY-MM-DD'),
       to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0)
Error report -
ORA-00001: unique constraint (KINO.SYS_C009852) violated
```

7.3. integralności referencji (czy wartości kluczy obcych są puste lub odpowiadają odpowiednim kluczom głównym z powiązanymi tabelami).

Sprawdzenie integralności referencji zostanie zaprezentowane na przykładzie dodawanych danych do tabeli Seances, która zawiera dwa klucze obce: id_movie oraz id_screening_room. W przypadku braku odpowiednich rekordów w tabelach: Movies oraz ScreeningRooms, przy próbie dodania danych do tabeli Seances wyświetlił się błąd.

Komenda:

```
INSERT INTO Seances (start_time, end_time, ticket_price, id_movie,
       id_screening_room)
VALUES (to_date('2022-02-01-12-00', 'YYYY-MM-DD-HH24-MI'),
       to_date('2022-02-01-14-50', 'YYYY-MM-DD-HH24-MI'), 15, 1, 2);
```

Komunikat:

```
Error starting at line : 626 in command -
INSERT INTO Movies (id_movie, title, length, distributor, borrow_date, return_date,
       dub_sub_lec, is3D)
VALUES (1,'Spiderman',120, 'xx distributor', to_date('2022-01-04', 'YYYY-MM-DD'),
       to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0)
Error report -
ORA-00001: unique constraint (KINO.SYS_C009852) violated
```

Po dodaniu odpowiednich rekordów do tabel Movies i ScreeningRooms oraz ponownej próbie dodania rekordu do tabeli Seances, w konsoli pojawił się komunikat o poprawnym dodaniu danych do tabeli Seances.

```
1 row inserted.
```

ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	22/02/01	22/02/01	15	1	2

7.4. testowanie procedur

Procedura insertResTeakenSeats:

```
CREATE OR REPLACE PROCEDURE insertResTeakenSeats(in_row_identifier char,
    in_column_identifier number,in_id_seance number, in_id_customer number)
AS
in_expiration_date date;
BEGIN
    SELECT start_time - 1/24 INTO in_expiration_date FROM SEANCES
    WHERE id_seance = in_id_seance;

    INSERT INTO TEAKENSEATS
    (EXPIRATION_DATE, ROW_IDENTIFIER, COLUMN_IDENTIFIER, RESERVED_OR_TEAKEN,
    ID_SEANCE, ID_TRANSACTION, ID_CUSTOMER)
    VALUES
    (in_expiration_date, in_row_identifier, in_column_identifier,
    'R', in_id_seance, NULL, in_id_customer);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertResTeakenSeats('A', 5, 1, 1);
EXECUTE insertResTeakenSeats('A', 6, 1, 1);
EXECUTE insertResTeakenSeats('B', 4, 2, 2);
```

Wynik działania procedury:

ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	1 22/02/01	A		5 R	1	(null)	1
2	2 22/02/01	A		6 R	1	(null)	1
3	3 22/02/01	B		4 R	2	(null)	2

Procedura insertTeakenSeats:

```
CREATE OR REPLACE PROCEDURE insertTeakenSeats(in_row_identifier char,
    in_column_identifier number,in_id_seance number, in_id_transaction number,
    in_id_customer number)
AS
in_expiration_date date;
BEGIN
    SELECT end_time + 2/24 INTO in_expiration_date FROM SEANCES
    WHERE id_seance = in_id_seance;

    INSERT INTO TEAKENSEATS
    (EXPIRATION_DATE, ROW_IDENTIFIER, COLUMN_IDENTIFIER, RESERVED_OR_TEAKEN,
    ID_SEANCE, ID_TRANSACTION, ID_CUSTOMER)
    VALUES
    (in_expiration_date, in_row_identifier, in_column_identifier,
    'T', in_id_seance, in_id_transaction, in_id_customer);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertTeakenSeats('A', 7, 1, NULL, NULL);
EXECUTE insertTeakenSeats('B', 2, 1, NULL, NULL);
EXECUTE insertTeakenSeats('B', 3, 1, NULL, NULL);
```

Wynik działania procedury:

4	4 22/02/01	A		7 T		1	(null)	(null)
5	5 22/02/01	B		2 T		1	(null)	(null)
6	6 22/02/01	B		3 T		1	(null)	(null)

Procedura insertTransactions:

```
CREATE OR REPLACE PROCEDURE insertTransactions(in_amount number,
    in_amount_of_tickets number,in_amount_of_reduced_tickets number,
    in_id_seance number, in_id_customer number)
AS
BEGIN
    INSERT INTO Transactions
    (amount, amount_of_tickets, amount_of_reduced_tickets, id_seance, id_customer)
    VALUES
    (in_amount, in_amount_of_tickets, in_amount_of_reduced_tickets,
    in_id_seance, in_id_customer);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertTransactions(30, 2, 0, 1, 1);
EXECUTE insertTransactions(20, 1, 0, 2, 2);
```

Wynik działania procedury:

	ID_TRANSACTION	AMOUNT	AMOUNT_OF_TICKETS	AMOUNT_OF_REDUCED_TICKETS	ID_SEANCE	ID_CUSTOMER
1	1	30	2	0	1	1
2	2	20	1	0	2	2

Procedura insertMovie:

```
CREATE OR REPLACE PROCEDURE insertMovie (in_title varchar2,in_length number,
                                         in_distributor varchar2,in_borrow_date date, in_return_date date,
                                         in_dub_sub_lec char,in_is3D number)
AS
BEGIN
    INSERT INTO Movies
    (title, length, distributor, borrow_date, return_date, dub_sub_lec, is3D)
    VALUES
    (in_title, in_length, in_distributor, in_borrow_date, in_return_date,
     in_dub_sub_lec, in_is3D);

    COMMIT;
END;
/
```

Wywołanie:

```
--DODAWANIE FILMU--
EXECUTE insertMovie('Superman',140, 'xx distributor',
to_date('2022-01-02', 'YYYY-MM-DD'), to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);
EXECUTE insertMovie('Spiderman',120, 'xx distributor',
to_date('2022-01-04', 'YYYY-MM-DD'), to_date('2022-02-04', 'YYYY-MM-DD'), 'D', 0);
SELECT * FROM Movies;
```

Wynik działania procedury:

	ID_MOVIE	TITLE	LENGTH	DISTRIBUTOR	BORROW_DATE	RETURN_DATE	DUB_SUB_LEC	IS3D
1	1	Superman	140	xx distributor	22/01/02	22/02/04	D	0
2	2	Spiderman	120	xx distributor	22/01/04	22/02/04	D	0

Procedura insertSeance:

```
CREATE OR REPLACE PROCEDURE insertSeance (in_start_time date, in_end_time date,
                                         in_ticket_price number, in_id_movie number, in_id_screening_room number)
AS
    movie_length number;
    data222 date;
BEGIN
    SELECT length INTO movie_length FROM movies WHERE id_movie = in_id_movie;

    IF in_end_time - in_start_time > movie_length/1440 THEN
        INSERT INTO Seances
        (start_time, end_time, ticket_price, id_movie, id_screening_room)
        VALUES
        (in_start_time, in_end_time, in_ticket_price, in_id_movie,
         in_id_screening_room);
    ELSE
        data222 := in_start_time + movie_length/1440;
        INSERT INTO Seances
        (start_time, end_time, ticket_price, id_movie, id_screening_room)
        VALUES
        (in_start_time, data222, in_ticket_price, in_id_movie,
         in_id_screening_room);
    END IF;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertSeance(to_date('2022-02-01-12-00', 'YYYY-MM-DD-HH24-MI'),
                      to_date('2022-02-01-14-50', 'YYYY-MM-DD-HH24-MI'), 15, 1, 2);
EXECUTE insertSeance(to_date('2022-02-01-15-00', 'YYYY-MM-DD-HH24-MI'),
                      to_date('2022-02-01-19-00', 'YYYY-MM-DD-HH24-MI'), 20, 2, 1);
```

Wynik działania procedury:

ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	22/02/01	22/02/01	15	1	2
2	22/02/01	22/02/01	20	2	1

Procedura insertScreeningRoom:

```
CREATE OR REPLACE PROCEDURE insertScreeningRoom (in_amount_of_rows number,
                                                in_amount_of_columns number)
AS
    in_amount_of_seats number;
BEGIN
    in_amount_of_seats := in_amount_of_rows*in_amount_of_columns;
    INSERT INTO ScreeningRooms
    (amount_of_seats, amount_of_rows, amount_of_columns)
    VALUES
    (in_amount_of_seats, in_amount_of_rows, in_amount_of_columns);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertScreeningRoom (8, 10);
EXECUTE insertScreeningRoom (7, 10);
EXECUTE insertScreeningRoom (8, 10);
```

Wynik działania procedury:

ID_SCREENING_ROOM	AMOUNT_OF_SEATS	AMOUNT_OF_ROWS	AMOUNT_OF_COLUMNS
1	1	80	8
2	2	70	7
3	3	80	8

Procedura insertCustomer:

```
CREATE OR REPLACE PROCEDURE insertCustomer (in_login varchar2,
    in_password varchar2, in_name varchar2,in_surname varchar2,
    in_phone_number varchar2)
AS
BEGIN
    INSERT INTO Customers
    (login, password, name, surname, phone_number, is_active)
    VALUES
    (in_login, in_password, in_name, in_surname, in_phone_number, 1);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertCustomer('sdsd4', 's3erf', 'Jan', 'Nowak', '758345903');
EXECUTE insertCustomer('8dsdf', '45rrrf', 'Pawel', 'Nowak', '958301903');
EXECUTE insertCustomer('555df', '4678rf', 'Tomasz', 'Nowak', '954561903');
```

Wynik działania procedury:

ID_CUSTOMER	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 sdsd4 s3erf		Jan	Nowak	758345903	1
2	2 8dsdf 45rrrf		Pawel	Nowak	958301903	1
3	3 555df 4678rf		Tomasz	Nowak	954561903	1

Procedura insertEmployee:

```
CREATE OR REPLACE PROCEDURE insertEmployee ( in_login varchar2,
    in_password varchar2, in_name varchar2, in_surname varchar2,
    in_phone_number varchar2, in_hire_date date, in_id_position number)
AS
BEGIN
    INSERT INTO Employees
    (login, password, name, surname, phone_number, hire_date, id_position)
    VALUES
    (in_login, in_password, in_name, in_surname, in_phone_number, in_hire_date,
    in_id_position);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertEmployee('empl1', 'sed3', 'Julia', 'Nowako', '657345903',
to_date('2020-01-02', 'YYYY-MM-DD'), 2);
EXECUTE insertEmployee('empl2', 'pwe4', 'Pawel', 'Towak', '923401903',
to_date('2021-06-02', 'YYYY-MM-DD'), 1);
EXECUTE insertEmployee('empl3', 'p554', 'Julian', 'Nowak', '567401903',
to_date('2021-06-02', 'YYYY-MM-DD'), 1);
```

Wynik działania procedury:

ID_EMPLOYEE	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	HIRE_DATE	FIRE_DATE	ID_POSITION
1	1 empl1 sed3		Julia	Nowako	657345903	20/01/02 (null)		2
2	2 empl2 pwe4		Pawel	Towak	923401903	21/06/02 (null)		1
3	3 empl3 p554		Julian	Nowak	567401903	21/06/02 (null)		1

Procedura insertPosition:

```
CREATE OR REPLACE PROCEDURE insertPosition (in_name varchar2, in_salary number)
AS
BEGIN
    INSERT INTO Positions (name, salary)
    VALUES (in_name, in_salary);

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE insertPosition('sprzedawca', 3000);
EXECUTE insertPosition('kierownik', 7000);
```

Wynik działania procedury:

ID_POSITION	NAME	SALARY
1	1 sprzedawca	3000
2	2 kierownik	7000

Procedura customerAccountDeactivation:

```
CREATE OR REPLACE PROCEDURE customerAccountDeactivation (in_id_customer number)
AS
BEGIN
    UPDATE Customers
    SET is_active = 0
    WHERE id_customer = in_id_customer;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE customerAccountDeactivation(2);
```

Wynik działania procedury:

ID_CUSTOMER	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 sdsd4 s3erf		Jan	Nowak	758345903	1
2	2 8dsdf 45rrrf		Pawel	Nowak	958301903	0
3	3 555df 4678rf		Tomasz	Nowak	954561903	1

Procedura changeCustomer:

```
CREATE OR REPLACE PROCEDURE changeCustomer (in_id_customer number,
    in_password varchar2, in_name varchar2, in_surname varchar2,
    in_phone_number varchar2)
AS
BEGIN
    UPDATE Customers
    SET password = in_password,
        name = in_name,
        surname = in_surname,
        phone_number = in_phone_number
    WHERE id_customer = in_id_customer;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE changeCustomer(3,'xxxxx', 'Juliusz', 'Slowacki', '888561903');
```

Wynik działania procedury:

ID_CUSTOMER	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 sdsd4 s3erf		Jan	Nowak	758345903	1
2	2 8dsdf 45rrrf		Pawel	Nowak	958301903	0
3	3 555df xxxxx		Juliusz	Slowacki	888561903	1

Procedura fireEmployee:

```
CREATE OR REPLACE PROCEDURE fireEmployee (in_id_employee number, in_fire_date date)
AS
BEGIN
    UPDATE Employees
    SET fire_date = in_fire_date
    WHERE id_employee = in_id_employee;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE fireEmployee(3, to_date('2022-01-02', 'YYYY-MM-DD'));
```

Wynik działania procedury:

ID_EMPLOYEE	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	HIRE_DATE	FIRE_DATE	ID_POSITION
1	1 empl1 sed3		Julia	Nowako	657345903	20/01/02 (null)		2
2	2 empl2 pwe4		Pawel	Towak	923401903	21/06/02 (null)		1
3	3 empl3 p554		Julian	Nowak	567401903	21/06/02	22/01/02	1

Procedura changePositionName:

```
CREATE OR REPLACE PROCEDURE changePositionName (in_id_position number, in_name varchar2)
AS
BEGIN
    UPDATE Positions
    SET name = in_name
    WHERE id_position = in_id_position;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE changePositionName(1, 'kier1');
```

Wynik działania procedury:

	ID_POSITION	NAME	SALARY
1	1	kier1	3000
2	2	kierownik	7000

Procedura changePositionSalary:

```
CREATE OR REPLACE PROCEDURE changePositionSalary (in_id_position number, in_salary number)
AS
BEGIN
    UPDATE Positions
    SET salary = in_salary
    WHERE id_position = in_id_position;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE changePositionSalary(1, 4550);
```

Wynik działania procedury:

	ID_POSITION	NAME	SALARY
1	1	kier1	4550
2	2	kierownik	7000

Procedura changeReservationToTeaken:

```
CREATE OR REPLACE PROCEDURE changeReservationToTeaken (in_id_seat number,
                                                       in_id_customer number)
AS
BEGIN
    UPDATE TeakenSeats
    SET reserved_or_teaken = 'T'
    WHERE id_seat = in_id_seat AND id_customer = in_id_customer;

    COMMIT;
END;
/
```

Wywołanie:

```
EXECUTE changeReservationToTeaken(3, 2);
```

Wynik działania procedury:

ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	1 22/02/01	A		5 R	1	(null)	1
2	2 22/02/01	A		6 R	1	(null)	1
3	3 22/02/01	B		4 T	2	(null)	2
4	4 22/02/01	A		7 T	1	(null)	(null)
5	5 22/02/01	B		2 T	1	(null)	(null)
6	6 22/02/01	B		3 T	1	(null)	(null)

Procedura deleteReservation:

```

CREATE OR REPLACE PROCEDURE deleteReservation (in_id_seat number,
                                             in_id_customer number)
AS
BEGIN
    DELETE FROM TeakenSeats
    WHERE (id_seat = in_id_seat AND id_customer = in_id_customer
          AND reserved_or_teaken = 'R');

    COMMIT;
END;
/

```

Wywołanie:

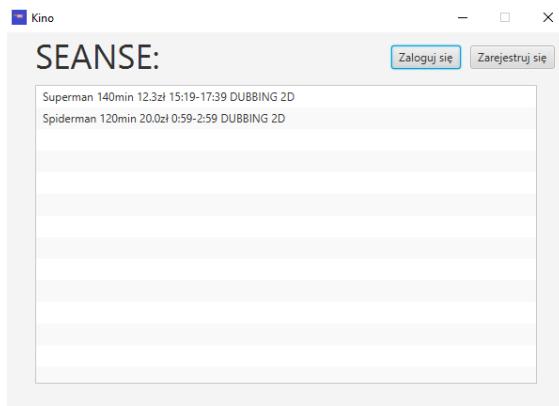
```
EXECUTE deleteReservation(2, 1);
```

Wynik działania procedury:

ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	1 22/02/01	A		5 R	1	(null)	1
2	3 22/02/01	B		4 T	2	(null)	2
3	4 22/02/01	A		7 T	1	(null)	(null)
4	5 22/02/01	B		2 T	1	(null)	(null)
5	6 22/02/01	B		3 T	1	(null)	(null)

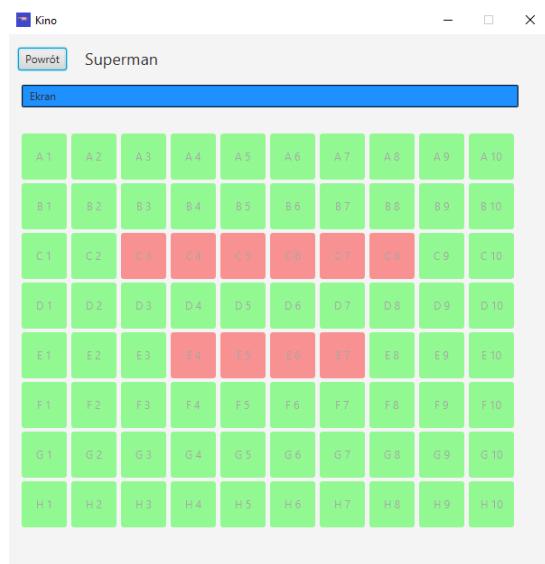
8. Makieta interfejsu graficznego aplikacji bazodanowej na podstawie scenariuszy funkcjonowania aplikacji bazodanowej i struktury bazy

Wszystkie okna w aplikacji nie są rozszerzalne.



Rysunek 5. Okno główne aplikacji

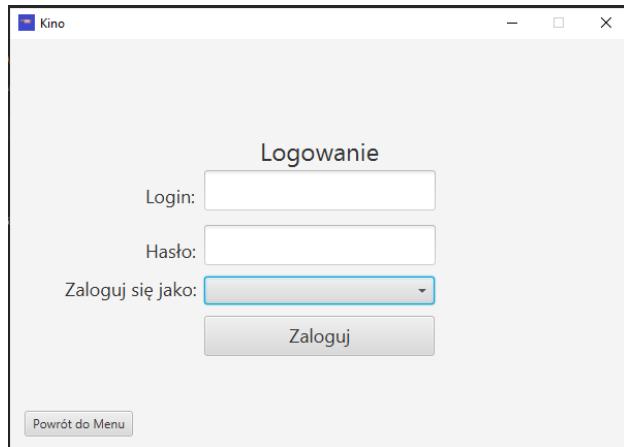
W oknie głównym aplikacji możemy zalogować się lub zarejestrować, bądź po kliknięciu na dany seans przejść do podglądu miejsc.



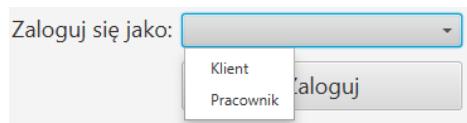
Rysunek 6. Okno Podglądu przykładowego seansu.

Siedzenia zaznaczone na: czerwono – zajęte, zielono – dostępne

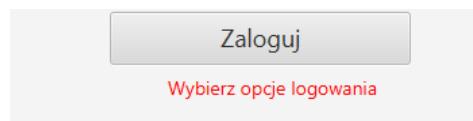
W oknie znajdują się wolne i zajęte miejsca przez innych użytkowników. Okno dostosowuje się wielkością do danej sali w której dzieje się seans.



Rysunek 7. Okno logowania

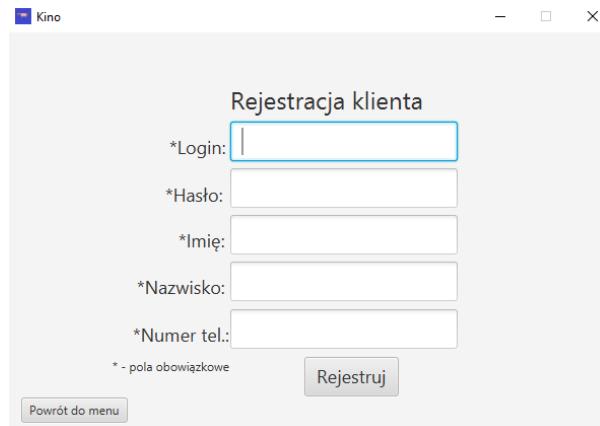


Rysunek 8. Wybór logowania



Rysunek 9. Przykładowy komunikat błędu dla okna logowania

W oknie logowania możemy zalogować się jako klient bądź pracownik. W razie podania błędnych danych pokaże się komunikat.

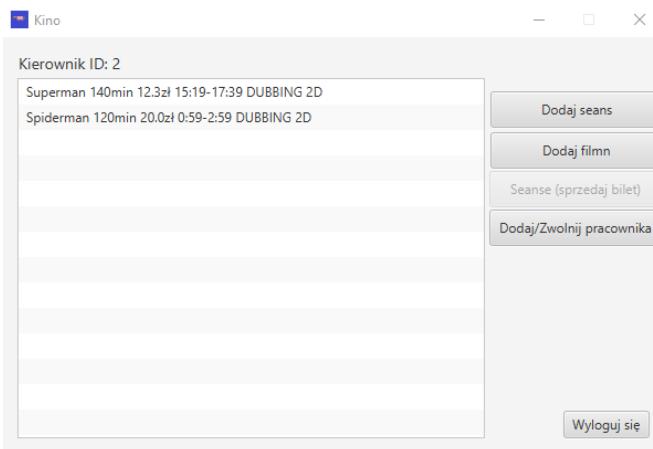


Rysunek 10. Okno rejestracji

Powyższe okno służy do rejestracji klientów. Aby się zarejestrować należy wpisać informacje w obowiązkowe pola. W razie błędnych danych, zajętego loginu bądź numeru telefonu klientowi pokaże się komunikat informujący o danym błędzie.

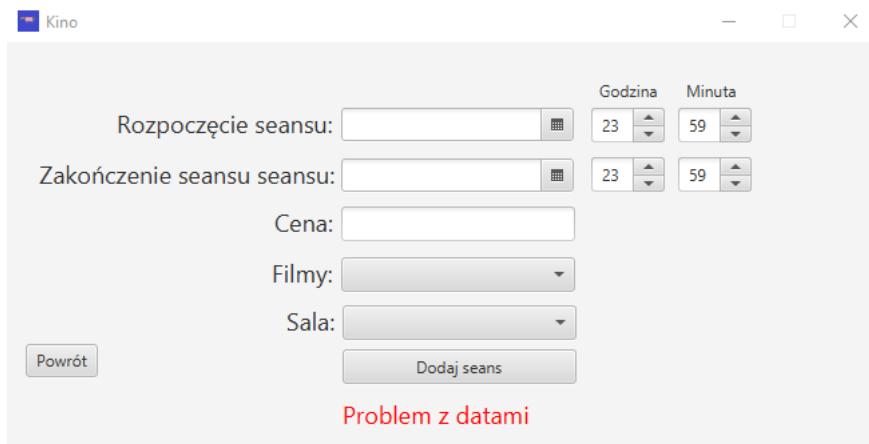


Rysunek 11. Przykładowy komunikat błędu dla okna rejestracji



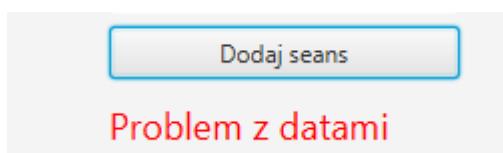
Rysunek 12. Okno główne pracownika

Okno pokazuje aktualne seanse. Aby móc sprzedać bilet trzeba wpierw wybrać seans a następnie nacisnąć guzik „Seanse (sprzedaj bilet)” (Bez wyboru sensu guzik będzie nie możliwy do naciśnięcia). Menu pozwala jeszcze przejść do okien dodawania filmów, seansów oraz jedynie po zalogowaniu jako kierownik dodaniu lub zwolnieniu pracownika, w innym przypadku guzik się nie pokaże.

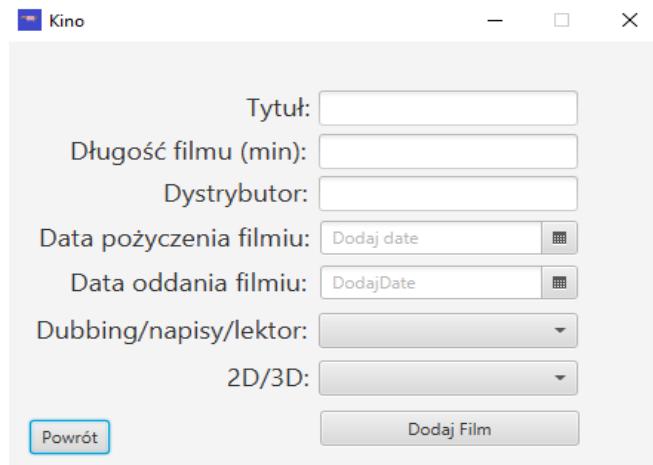


Rysunek 13. Okno dodania filmu

Aby dodać seans należy uzupełnić wszystkie informacje. W razie podania złych informacji takich jak np.: długość filmu dłuższa niż długość seansu, zaplanowanie sensu w złych godzinach bądź dla filmu, który będzie już oddany itp. ukaże się komunikat informacyjny o popełnionym błędzie.



Rysunek 14. Przykładowy komunikat błędu dla okna dodania seansu

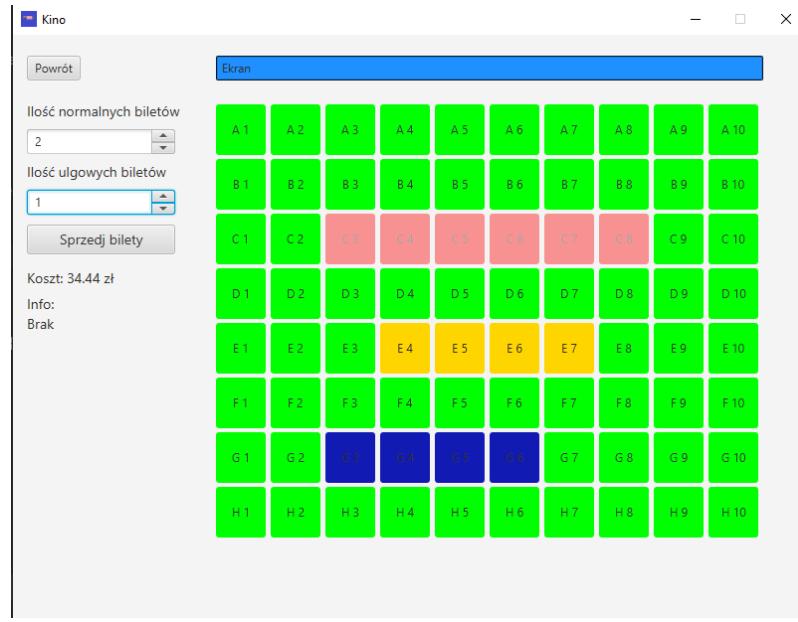


Rysunek 15. Okno dodania filmu

Aby dodać film należy uzupełnić wszystkie informacje. W razie podania złych informacji pojawi się komunikat błędu.



Rysunek 16. Przykładowy komunikat błędu dla okna dodania filmu

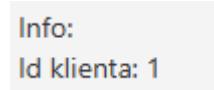


Rysunek 17. Okno sprzedaży biletów

Siedzenia zaznaczone na:

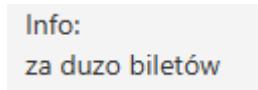
czerwono – zajęte, żółto – zarezerwowane, niebiesko – aktualnie wybrane, zielono – dostępne.

Aby sprzedać bilet należy wybrać miejsca oraz wybrać ilość biletów a następnie nacisnąć guzik „Sprzedaj bilety”. W razie wyboru miejsca zarezerwowanego pokaże się informacja z id klienta który dokonał rezerwacji.

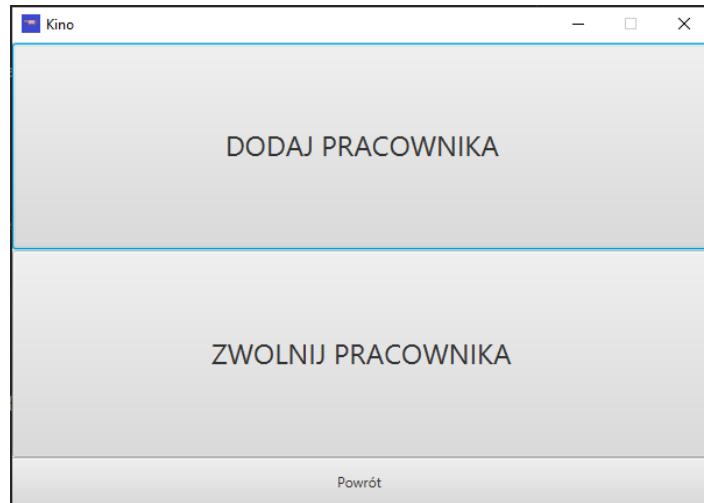


Rysunek 18. Informacja o id klienta

W razie wybrania złej liczby biletów pokaże się komunikat.



Rysunek 19. Przykładowy komunikat błędu dla okna sprzedaży biletów



Rysunek 20. Okno edycji pracowników

Gdy zalogowaliśmy się do aplikacji jako kierownik mamy prawo do zwalniania bądź dodawania pracowników.

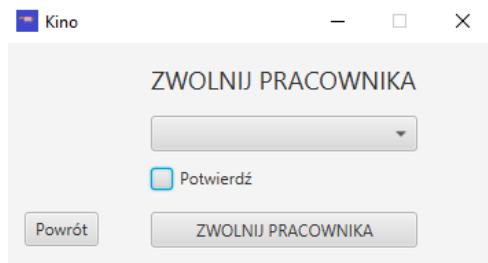
A screenshot of a registration form titled 'Rejestracja pracownika'. The form includes fields for 'Login', 'Hasło', 'Imię', 'Nazwisko', 'Numer tel.', and 'Pozycja'. There is also a 'Rejestruj' button and a 'Powrót do menu' button. A note at the top left indicates that fields marked with an asterisk (*) are mandatory.

Rysunek 21. Okno dodawania pracowników

Aby dodać pracownika należy wypełnić informacje a następnie przycisnąć guzik „Rejestruj”. W razie błędnych informacji pokaże się komunikat.

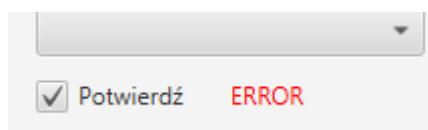


Rysunek 22. Przykładowy komunikat błędu dla okna dodania pracownika

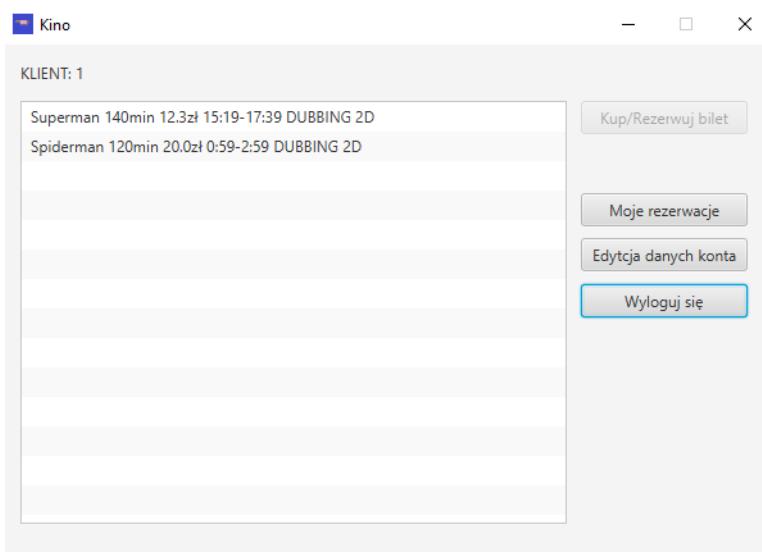


Rysunek 23. Okno zwolnienia pracownika

Aby zwolnić pracownika należy wybrać pracownika z listy pracowników, potwierdzić a następnie nacisnąć guzik „ZWOLNIJ PRACOWNIKA”. W problemie pokaże się komunikat.

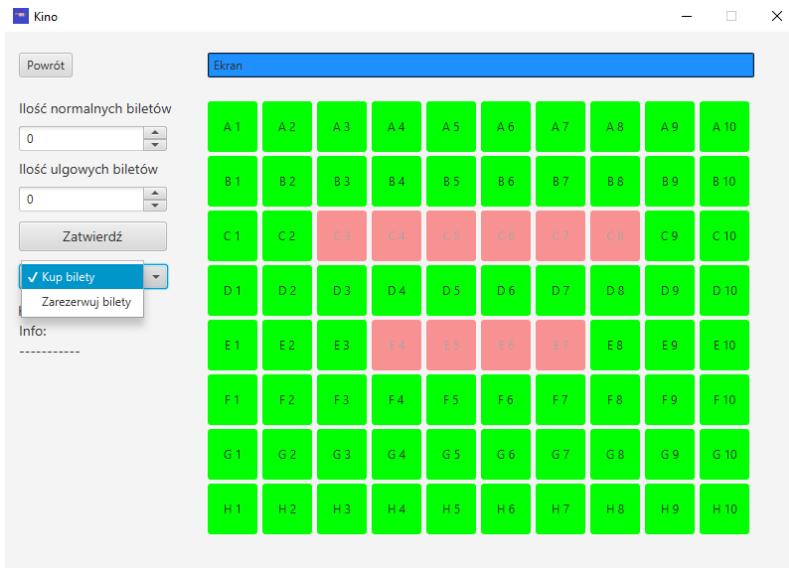


Rysunek 24. Przykładowy komunikat błędu dla okna zwolnienia pracownika



Rysunek 25. Okno główne klienta

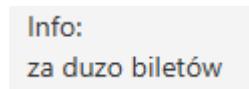
Okno pokazuje aktualne seanse. Aby móc kupić/zarezerwować bilet trzeba wpierw wybrać seans a następnie nacisnąć guzik „Kup/Rezerwuj bilet” (Bez wyboru sensu guzik będzie nie możliwy do naciśnięcia). Menu posiada również opcje przejścia do okien z rezerwacjami, kupionymi biletami oraz edycją konta.



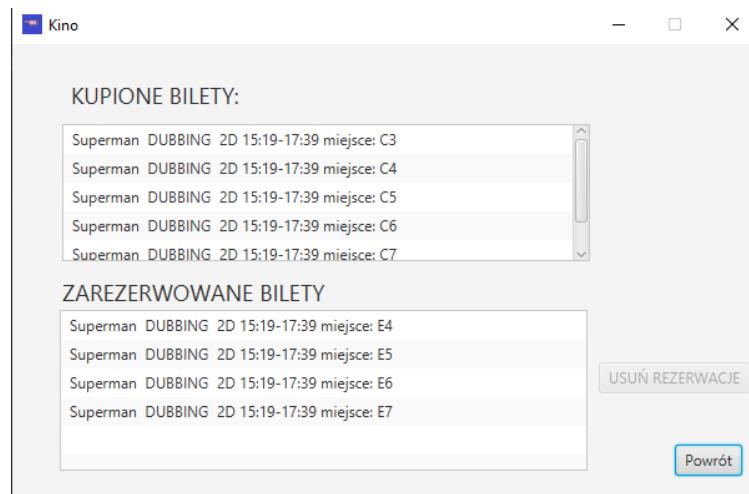
Rysunek 26. Okno kupna/rezerwacji biletów

Aby kupić/zarezerwować bilet należy wybrać miejsca oraz ilość biletów, opcje czy kupujemy bądź rezerwujemy a następnie nacisnąć guzik „Zatwierdź”.

W razie wybrania złej liczby biletów pokaże się komunikat.

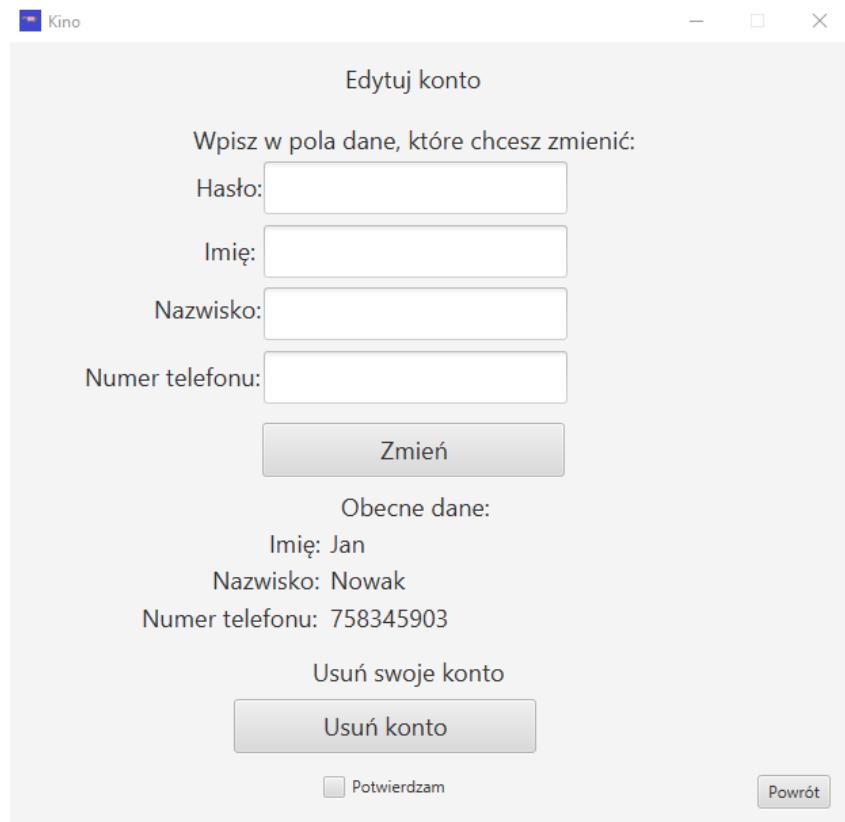


Rysunek 27. Przykładowy komunikat błędu dla okna kupna/rezerwacji biletów



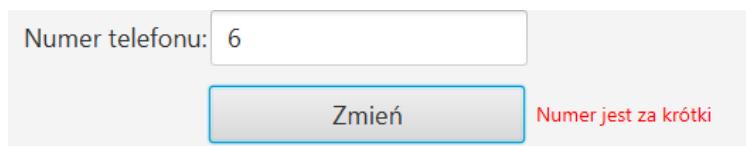
Rysunek 28. Okno rezerwacje klienta

Okno pokazuje Kupione oraz zarezerwowane bilety dla danego klienta. Można usunąć rezerwacje wybierając na dany bilet z listy „ZAREZERWOWANE BILETY” a następnie przyciskając guzik „USUŃ REZERWACJĘ”.

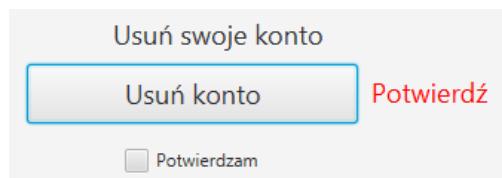


Rysunek 29. Okno edycji konta klienta

Okno pozwala na edycję danych oraz jego usunięcie. W razie błędnych danych bądź nie błędu przy usuwaniu konta pojawią się odpowiednie komunikaty.

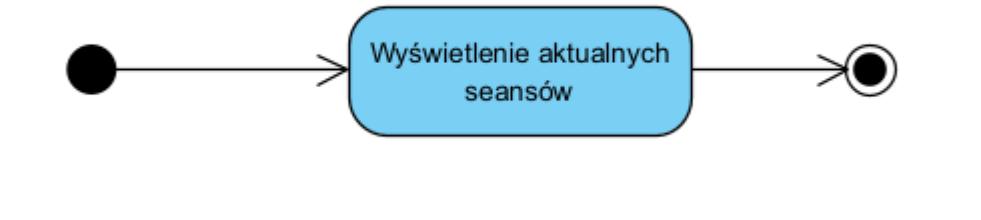


Rysunek 30. Przykładowy komunikat błędu dla zmiany danych klienta

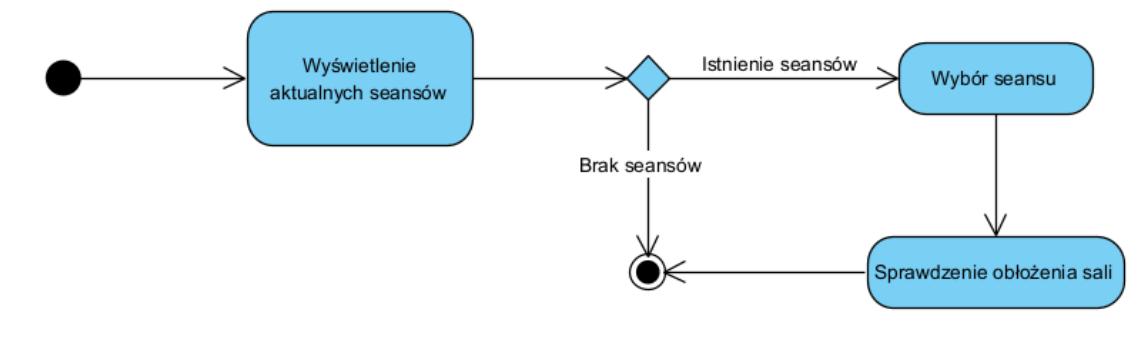


Rysunek 31. Przykładowy komunikat błędu dla usunięcia konta klienta

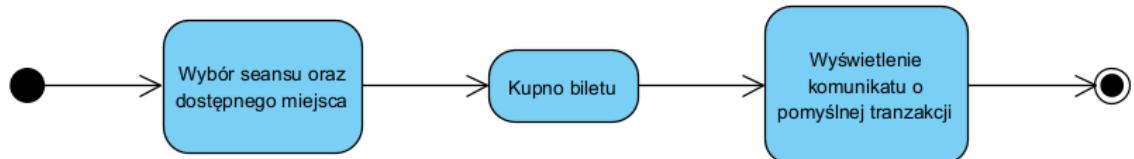
9. Diagramy czynności dla przypadków użycia: algorytm funkcjonowania aplikacji bazodanowej



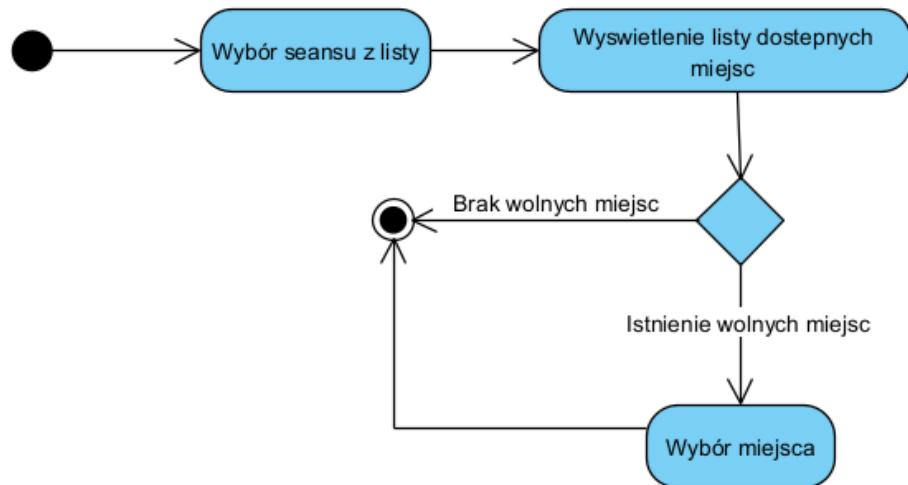
Rysunek 32. PU Sprawdzenie listy seansów



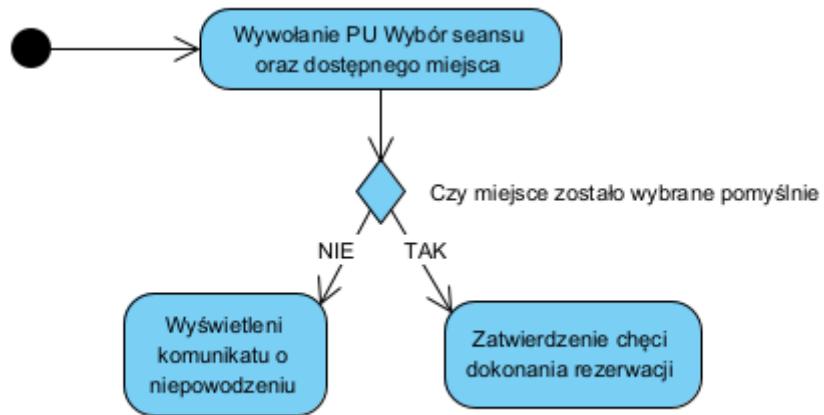
Rysunek 33. PU Sprawdzenie obłożenia Sali



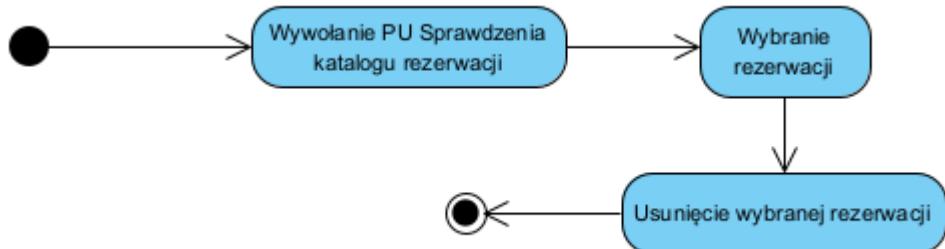
Rysunek 34. PU Kupno biletu



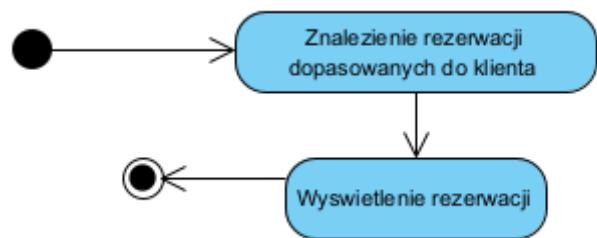
Rysunek 35. PU Wybór seansu oraz dostępnego miejsca



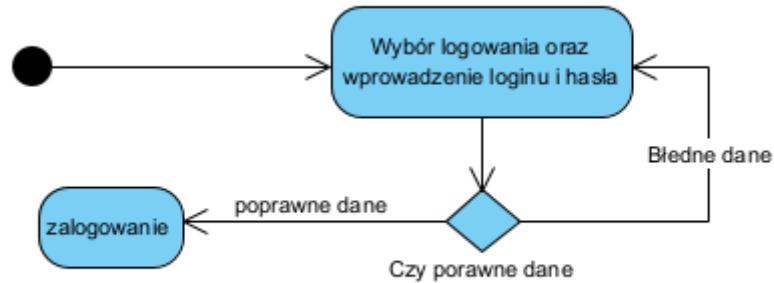
Rysunek 36. PU Rezerwacja biletu



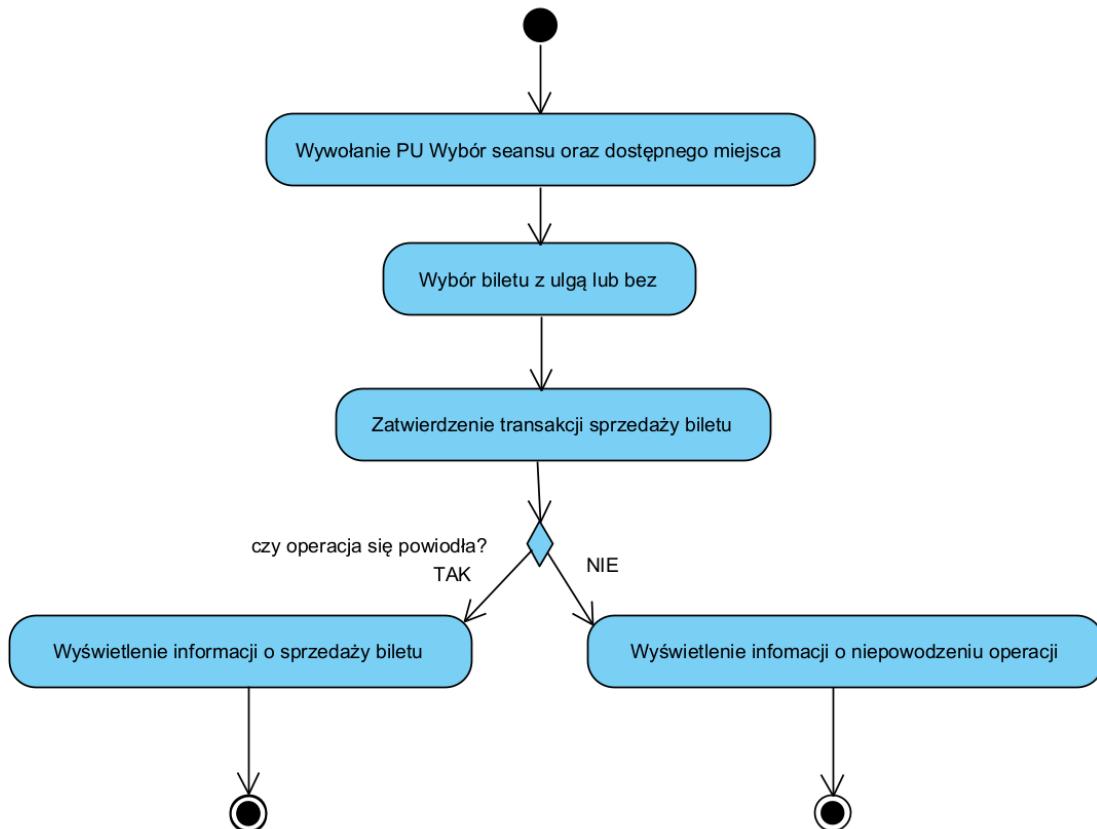
Rysunek 37. PU Odstąpienie od rezerwacji



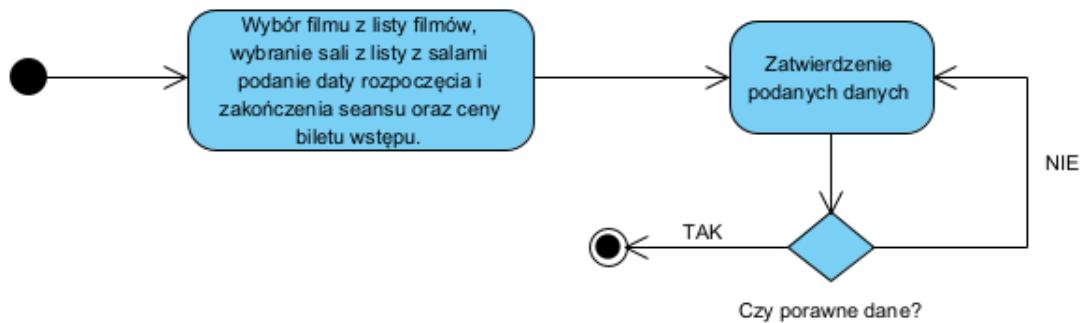
Rysunek 38. PU Sprawdzenia katalogu rezerwacji



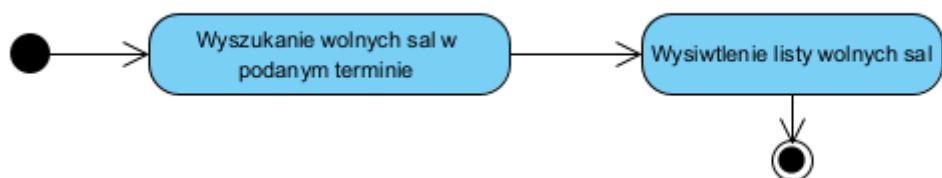
Rysunek 39. PU Logowanie do systemu



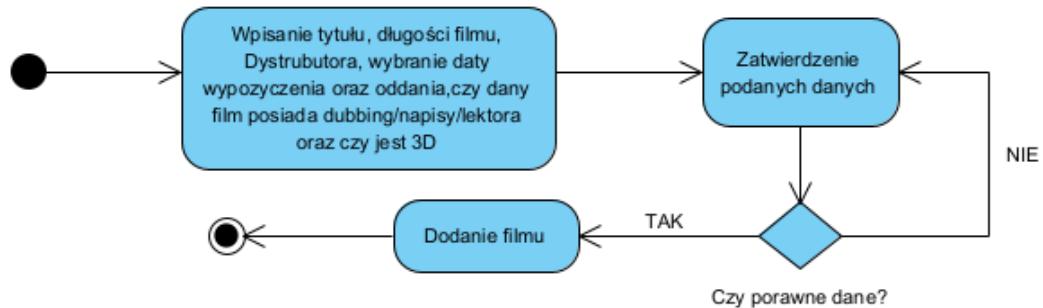
Rysunek 40. PU Sprzedaż biletów



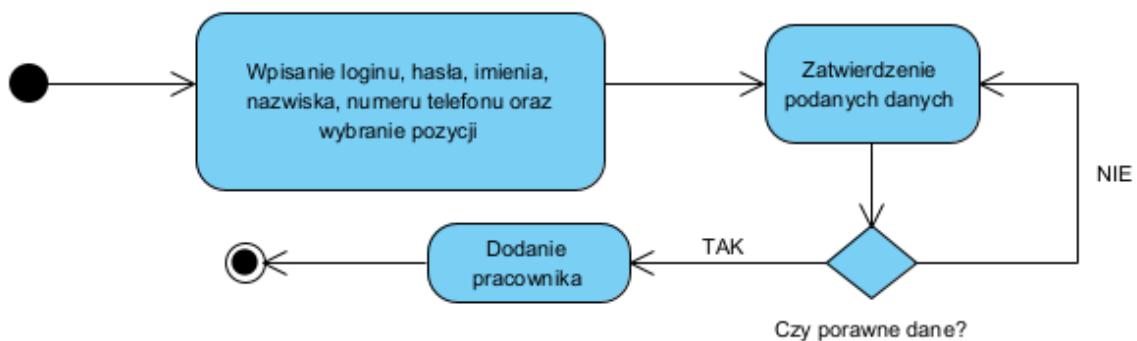
Rysunek 41. PU Tworzenie seansów



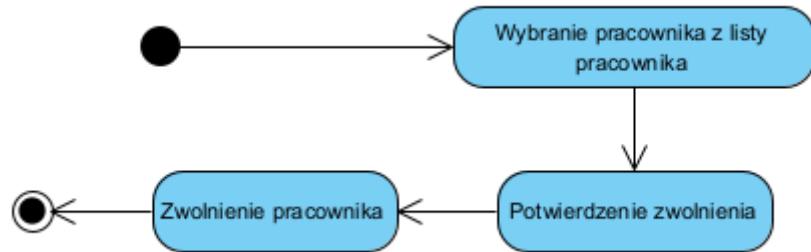
Rysunek 42. PU Sprawdzenie dostępności Sali



Rysunek 43. PU Dodawanie filmów



Rysunek 44. PU Zatrudnianie pracowników

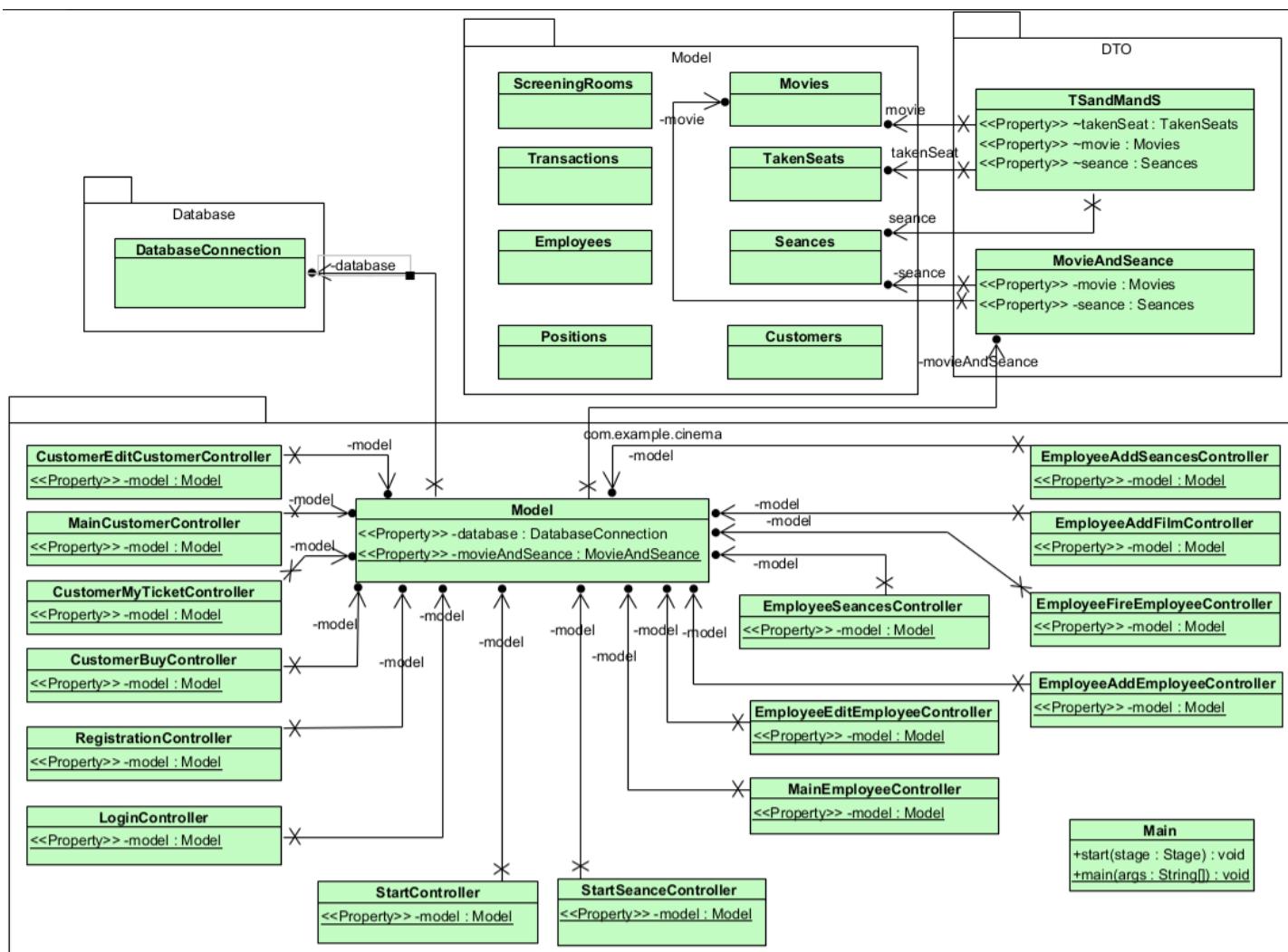


Rysunek 45. PU Zwalnianie pracowników

Przypadek użycia modyfikacja danych pracowników oraz edycja stanowisk pracowników nie zostały dodane do projektu. Było to spowodowane brakiem czasu.

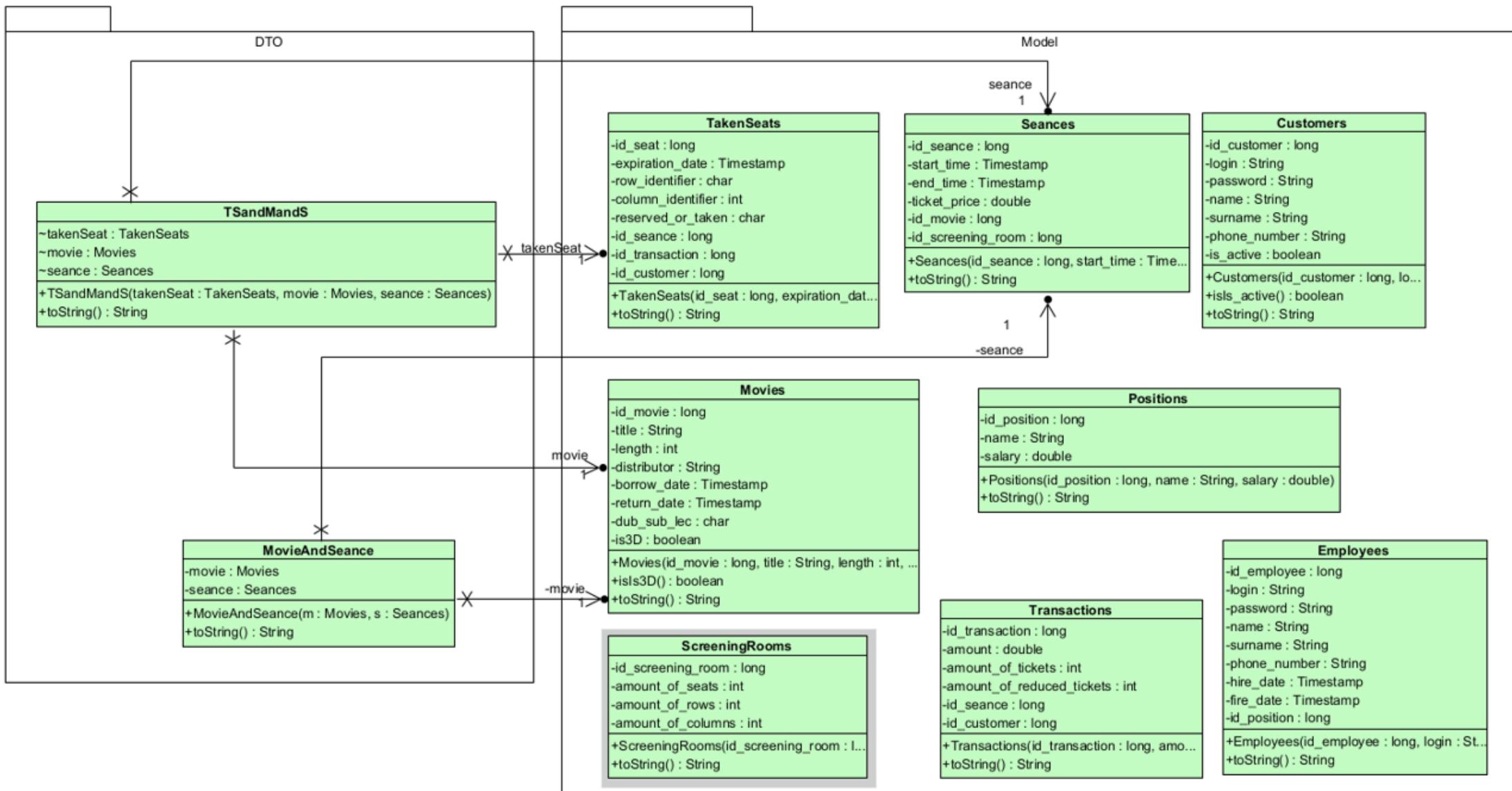
10. Diagram klas i jego powiązania w aplikacji bazodanowej

10.1. Wszystkie klasy

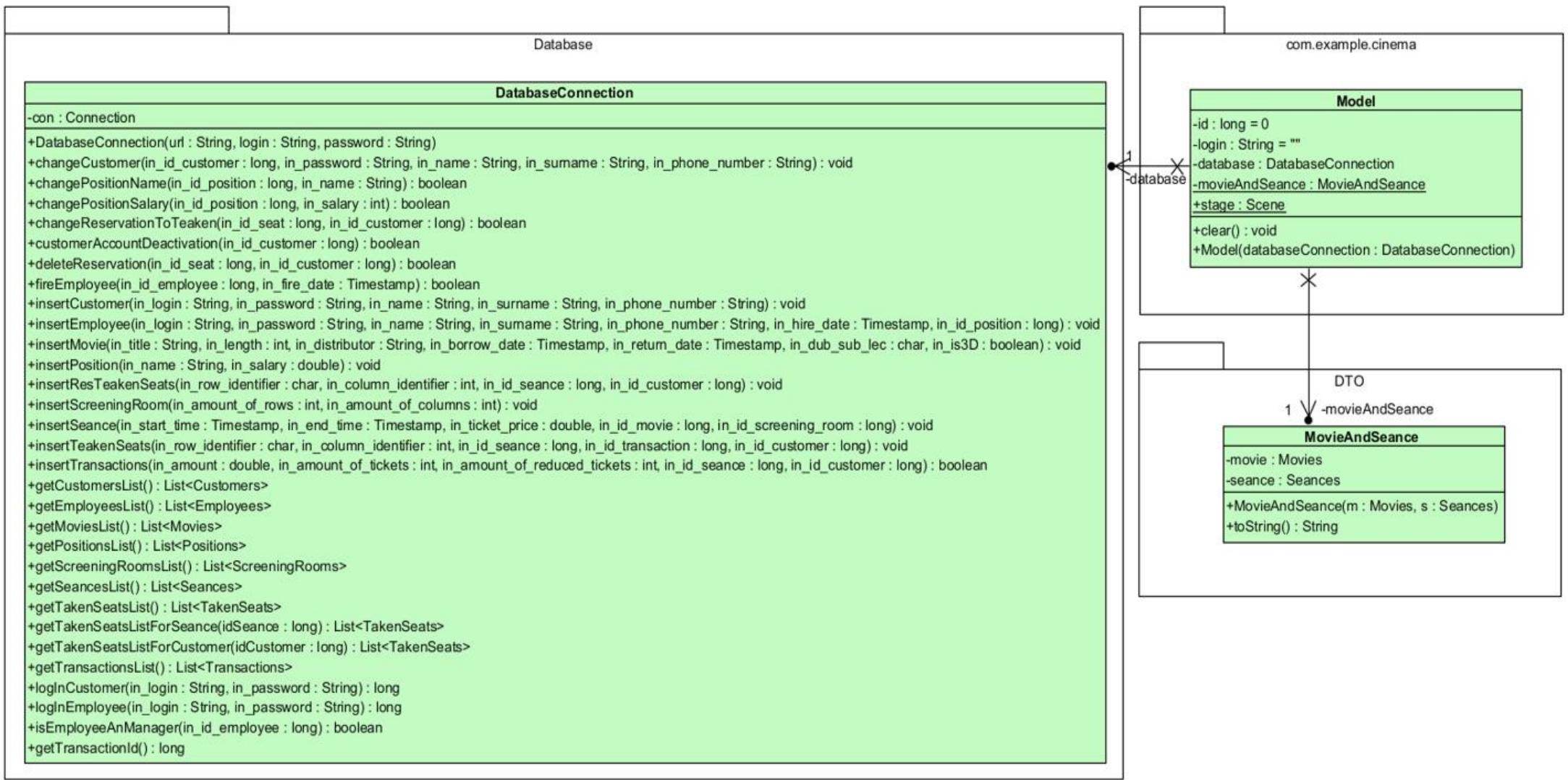


Rysunek 46. Cały diagram klas

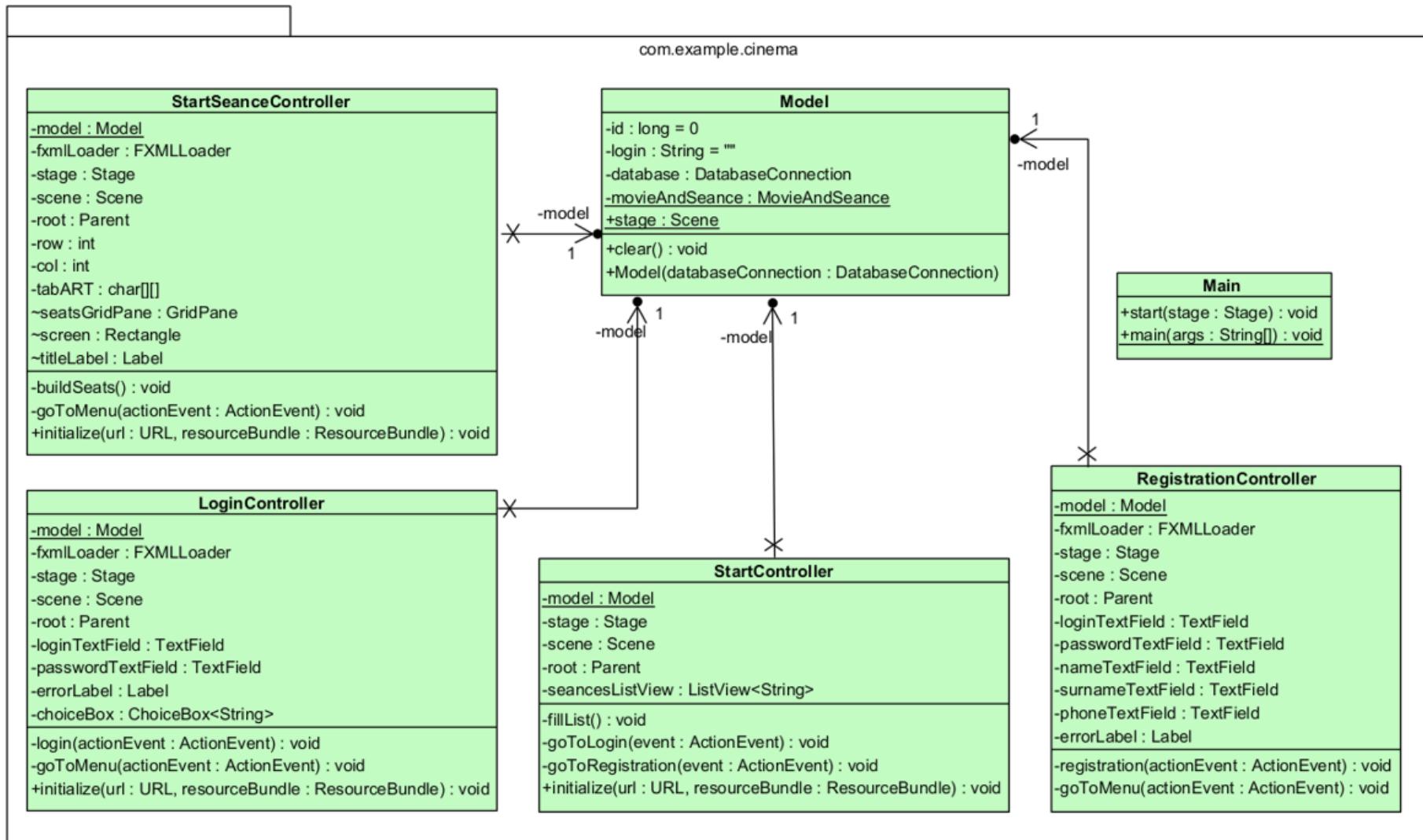
10.2. Diagramy



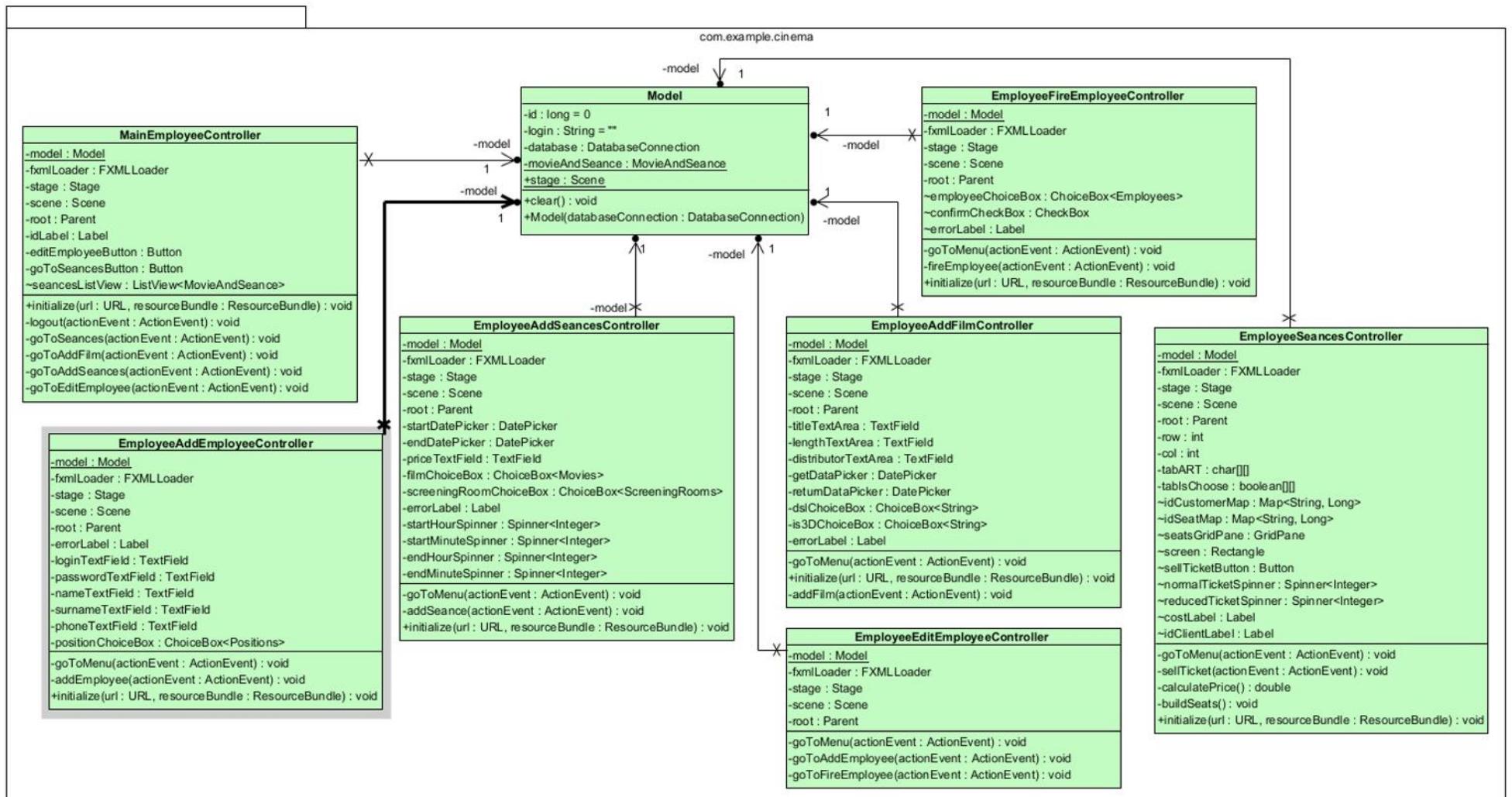
Rysunek 47. Diagram klas dla paczek DTO i Model



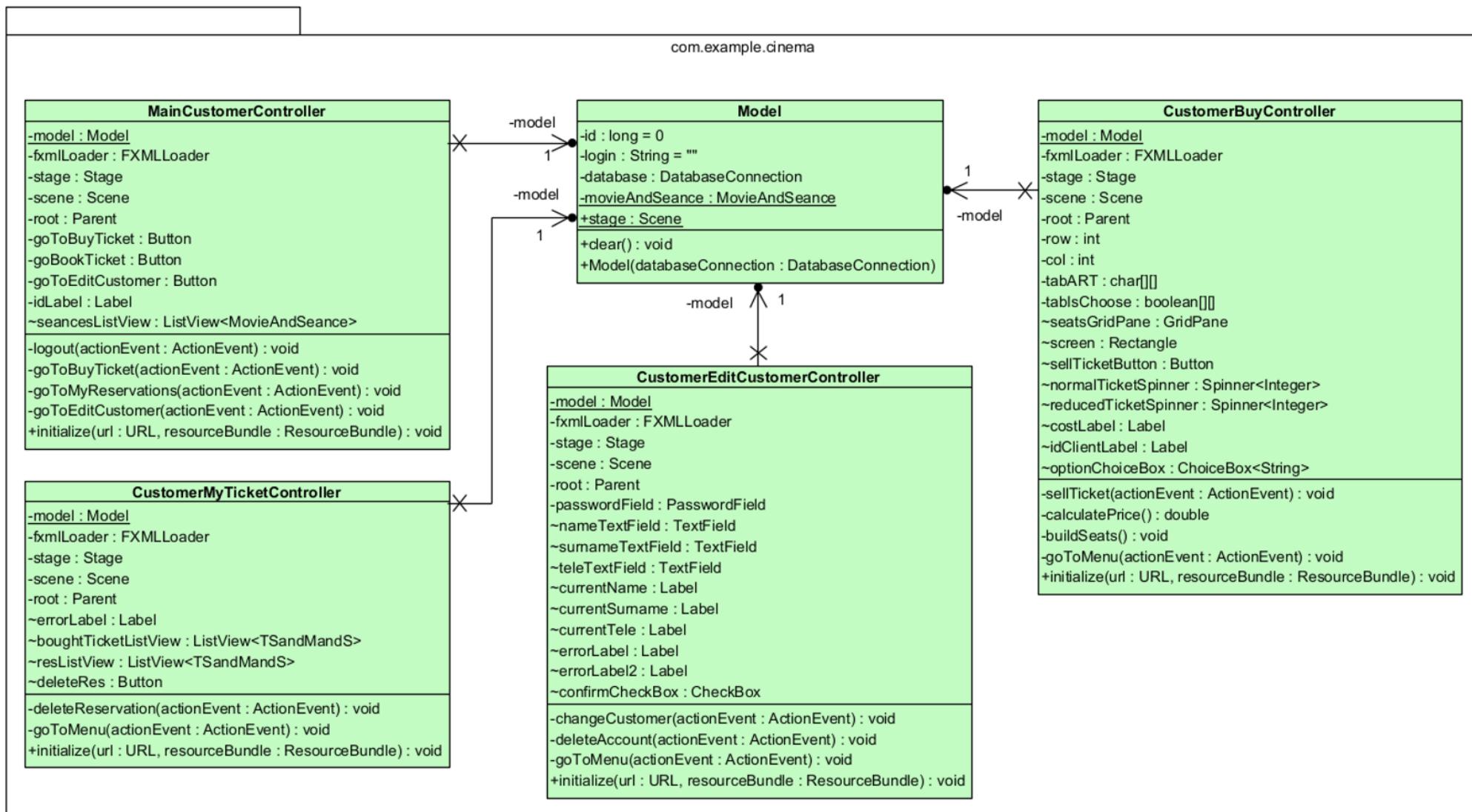
Rysunek 48. Diagram klas dla DatabaseConnection i Model



Rysunek 49. Diagram klas dla wspólnych kontrolerów



Rysunek 50. Diagram klas dla kontrolerów pracownika



Rysunek 51. Diagram klas dla kontrolerów klienta

11. Testy

11.1. Testy ręczne

11.2. Logowanie (LoginController, LoginView.fxml)

Dane z tabeli CUSTOMERS:

ID_CUSTOMER	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 Natalia	Natalia	Jan	Nowak	758345903	1
2	2 Login66	Haslo66	Pawel	Nowak	958301903	0
3	3 Matryca	Matryca	Juliusz	Slowacki	888561903	1
4	5 Klient	Klient	Pawel	Trybek	223456111	1

Dane z tabeli EMPLOYEES:

ID_EMPLOYEE	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	HIRE_DATE	FIRE_DATE	ID_POSITION
1	5 Kierownik	Kierownik	Michał Jank...	603446881	23/01/27 (null)			2
2	1 123456	123456	Julixa Noxwako	65734903	22/02/01 (null)			2
3	3 Kasia5	Kasia12345	Pawel Towak	923401903	22/02/01	22/02/01		1
4	4 MartaKK	p55aaaaa4	Julian Nowak	567401903	22/02/01	(null)		1

Wpiswanie i wybieranie błędnych danych:

Logowanie

Login:

Hasło:

Zaloguj się jako:

Wybierz opcje logowania

Logowanie

Login:

Hasło:

Zaloguj się jako:

Błąd

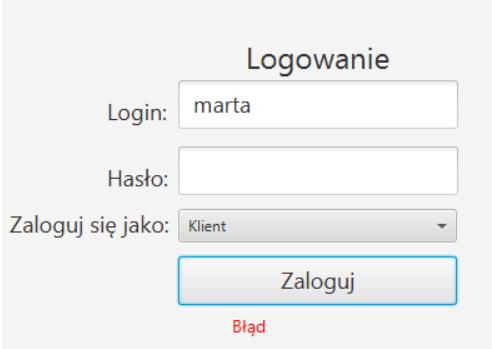
Logowanie

Login:

Hasło:

Zaloguj się jako:

Błąd



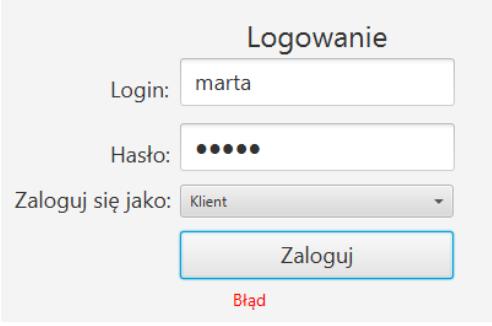
Logowanie

Login:

Hasło:

Zaloguj się jako:

Błąd



Wpisanie danych kierownik, ale wybór logowania jako klient

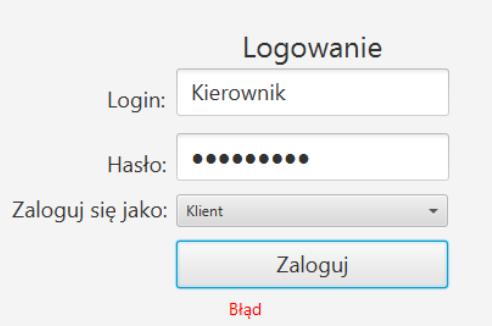
Logowanie

Login:

Hasło:

Zaloguj się jako:

Błąd



Logowanie się na uśnięte konto klienta

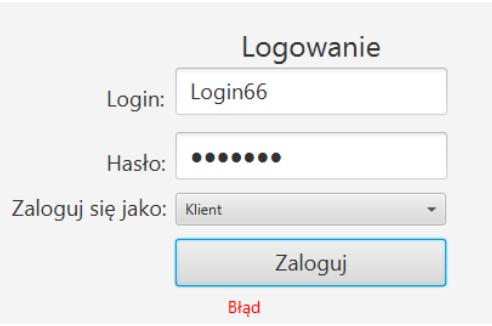
Logowanie

Login:

Hasło:

Zaloguj się jako:

Błąd



Logowanie się na konto zwolnionego pracownika

Logowanie

Login: Kasia5

Hasło: ██████████

Zaloguj się jako: Pracownik ▾

Zaloguj

Błąd

Wpiswanie i wybieranie poprawnych danych:

Wpisanie loginu i hasła „Kierownik” (logowanie jako kierownik)

<h2>Logowanie</h2>	
Login:	Kierownik
Hasło:	••••••••
Zaloguj się jako:	Pracownik ▾
Zaloguj	

Spowodowało zalogowanie się jako kierownik

Kierownik ID: 5	
Superman 140min 12.3zł 15:19-17:39 DUBBING 2D	Dodaj seans
Spiderman 120min 20.0zł 0:59-2:59 DUBBING 2D	Dodaj filmn
	Seanse (sprzedaj bilet)
	Dodaj/Zwolnij pracownika
	Wyloguj się

Wpisanie loginu „MartaKK” i hasła „p55aaaa4” (logowanie jako pracownik)

Logowanie

Login:	MartaKK
Hasło:	••••••••••
Zaloguj się jako:	Pracownik ▾
Zaloguj	

Spowodowało zalogowanie się jako pracownik

Pracownik ID: 4	
Superman 140min 12.3zł 15:19-17:39 DUBBING 2D	Dodaj seans
Spiderman 120min 20.0zł 0:59-2:59 DUBBING 2D	Dodaj filmn
	Seanse (sprzedaj bilet)
	Wyloguj się

11.3. Rejestracja (RegistrationController, RegistrationView.fxml)

Dane z tabel CUSTOMERS i EMPLOYEES są takie same jak dla punktu a)

Wpiswanie i wybieranie błędnych danych:

Login jest za krótki

Rejestracja klienta

*Login:	<input type="text"/>
*Hasło:	<input type="password"/>
*Imię:	<input type="text"/>
*Nazwisko:	<input type="text"/>
*Numer tel.:	<input type="text"/>

* - pola obowiązkowe

Hasło jest za krótkie	
Rejestracja klienta	
*Login:	karta33
*Hasło:	
*Imię:	
*Nazwisko:	
*Numer tel.:	
- pola obowiązkowe	
Rejestruj	

ZŁY NUMER

Rejestracja klienta

*Login: karta33

*Hasło: ████████

*Imię:

*Nazwisko:

*Numer tel.:

* - pola obowiązkowe

Rejestruj

ERROR

Rejestracja klienta

*Login: karta33

*Hasło: ████████

*Imię:

*Nazwisko:

*Numer tel.: 222333444

* - pola obowiązkowe

Rejestruj

ERROR

Rejestracja klienta

*Login: karta33

*Hasło: ████████

*Imię: Maja

*Nazwisko:

*Numer tel.: 222333444

* - pola obowiązkowe

Rejestruj

Wpiswanie i wybieranie poprawnych danych:

Rejestracja klienta

*Login: karta33

*Hasło: ████████

*Imię: Maja

*Nazwisko: Pawlikow

*Numer tel.: 222333444

* - pola obowiązkowe

Rejestruj

Tabel CUSTOMERS po wszystkich interakcjach:

ID_CUST...	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 Natalia	Natalia	Jan	Nowak	758345903	1
2	2 Login66	Haslo66	Pawel	Nowak	958301903	0
3	3 Matryca	Matryca	Juliusz	Slowacki	888561903	1
4	5 Klient	Klient	Paweł	Trybek	223456111	1
5	9 karta33	karta33	Maja	Pawlikow	222333444	1

11.4. Głównie okno pracownika (MainEmployeeController, MainEmployeeView.fxml)

Brak możliwości ręcznych testów

11.5. Dodawanie seansu (EmployeeAddSeancesController, EmployeeAddSeancesView.fxml)

Tabela z danymi SEANCES:

ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	1 22/02/01	22/02/01	15,3	1	2
2	2 22/02/01	22/02/01	20,38	2	1
3	3 23/01/28	23/01/28	12,3	1	1
4	4 23/01/28	23/01/28	20	2	3

Tabela z danymi MOVIES:

ID_MOVIE	TITLE	LENGTH	DISTRIBUTOR	BORROW_DATE	RETURN_DATE	DUB_SUB_LEC	IS3D
1	3 Batman	180	WB	23/01/29	23/01/31	S	1
2	1 Superman	140	xx distributor	23/02/01	22/03/01	D	0
3	2 Spiderman	120	xx distributor	23/02/01	22/03/01	D	0

Wpiswanie i wybieranie błędnych danych:

Rozpoczęcie seansu: Godzina Minuta

Zakończenie seansu seansu:

Cena:

Filmy:

Sala:

Powrót Dodaj seans

Problem z datami

Kino

Rozpoczęcie seansu: 28.01.2023 Godzina 15 Minuta 59

Zakończenie seansu seansu: 28.01.2023 17 59

Cena:

Filmy:

Sala:

Powrót Dodaj seans

Problem ceną

Rozpoczęcie seansu: 27.01.2023 Godzina 23 Minuta 59

Zakończenie seansu seansu: 27.01.2023 21 59

Cena: 20.3

Filmy: id_movie=1, title=Superma...

Sala: id_screening_room=1, amo...

Powrót Dodaj seans

Zły data lub czas

Kino

Rozpoczęcie seansu: 28.01.2023 Godzina 15 Minuta 59

Zakończenie seansu seansu: 28.01.2023 17 59

Cena: ssss

Filmy:

Sala:

Powrót Dodaj seans

Problem ceną

Film jest dłuższy od seansu

Kino

Rozpoczęcie seansu: 28.01.2023 Godzina 15 Minuta 59

Zakończenie seansu seansu: 28.01.2023 16 59

Cena: 20.3

Filmy: id_movie=1, title=Superma...

Sala: id_screening_room=1, amo...

Powrót Dodaj seans

Zły data lub czas

Rozpoczęcie seansu: 28.01.2023 Godzina 23 Minuta 59

Zakończenie seansu seansu: 29.01.2023 23 59

Cena: 22

Filmy:

Sala:

Powrót Dodaj seans

Nie wybrano filmu bądź sali

Rozpoczęcie seansu: 5.02.2023 Godzina 12 Minuta 59

Zakończenie seansu seansu: 5.02.2023 16 59

Cena: 22

Filmy: id_movie=3, title=Batman, I...

Sala: id_screening_room=2, amo...

Powrót Dodaj seans

FILM JUZ ZOSTANIE ODDANY

Rozpoczęcie seansu: Godzina Minuta

Zakończenie seansu seansu:

Cena:

Filmy:

Sala:

FILM JESZCZE NIE BEDZIE WYPOZYCZONY

Wpiswanie i wybieranie poprawnych danych:

Rozpoczęcie seansu: Godzina Minuta

Zakończenie seansu seansu:

Cena:

Filmy:

Sala:

DODANO SEANS

Tabel SEANCES po wszystkich interakcjach:

ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	23/01/29	23/01/29	22	3	2
2	22/02/01	22/02/01	15,3	1	2
3	22/02/01	22/02/01	20,38	2	1
4	23/01/28	23/01/28	12,3	1	1
5	23/01/28	23/01/28	20	2	3

11.6. Dodawanie filmu (EmployeeAddFilmController, EmployeeAddFilmView.fxml)

Tabela z danymi MOVIES:

ID_MOVIE	TITLE	LENGTH	DISTRIBUTOR	BORROW_DATE	RETURN_DATE	DUB_SUB_LEC	IS3D
1	3 Batman	180	WB	23/01/29	23/01/31	S	1
2	1 Superman	140	xx distributor	23/02/01	22/03/01	D	0
3	2 Spiderman	120	xx distributor	23/02/01	22/03/01	D	0

Wpiswanie i wybieranie błędnych danych:

 Kino

Brak nazwy filmu

Tytuł:

Długość filmu (min):

Dystrybutor:

Data pożyczenia filmu: Dodaj date

Data oddania filmu: DodajDate

Dubbing/napisy/lektor:

2D/3D:

Zła długość filmu

Tytuł: Hulk

Długość filmu (min):

Dystrybutor:

Data pożyczenia filmu: Dodaj date

Data oddania filmu: DodajDate

Dubbing/napisy/lektor:

2D/3D:

Problem z datami

Tytuł: Hulk

Długość filmu (min): 110

Dystrybutor: WB

Data pożyczenia filmu: Dodaj date

Data oddania filmu: DodajDate

Dubbing/napisy/lektor:

2D/3D:

Nie wybrano dub/nap/lek

Tytuł:	Hulk
Długość filmu (min):	110
Dystrybutor:	WB
Data pożyczania filmu:	28.01.2023
Data oddania filmu:	29.01.2023
Dubbing/napisy/lektor:	▼
2D/3D:	▼

Powrót Dodaj Film

Nie wybrano 2D/3D

Tytuł:	Hulk
Długość filmu (min):	110
Dystrybutor:	WB
Data pożyczania filmu:	28.01.2023
Data oddania filmu:	29.01.2023
Dubbing/napisy/lektor:	Dubbing
2D/3D:	▼

Powrót Dodaj Film

Wpiswanie i wybieranie poprawnych danych:

Kino

DODANO FILM

Tytuł:	Hulk
Długość filmu (min):	110
Dystrybutor:	WB
Data pożyczania filmu:	28.01.2023
Data oddania filmu:	29.01.2023
Dubbing/napisy/lektor:	Napisy
2D/3D:	3D

Powrót Dodaj Film

Tabel MOVIES po wszystkich interakcjach:

	ID_MOVIE	TITLE	LENGTH	DISTRIBUTOR	BORROW_DATE	RETURN_DATE	DUB_SUB_LEC	IS3D
1	3	Batman	180	WB	23/01/29	23/01/31	S	1
2	4	Hulk	110	WB	23/01/28	23/01/29	S	1
3	1	Superman	140	xx distributor	23/02/01	22/03/01	D	0
4	2	Spiderman	120	xx distributor	23/02/01	22/03/01	D	0

11.7. Sprzedaż biletów (EmployeeSeancesController, EmployeeSeancesView.fxml)

Tabela z danymi SEANCES:

	ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	8	23/01/29	23/01/29	22	3	2
2	1	22/02/01	22/02/01	15,3	1	2
3	2	22/02/01	22/02/01	20,38	2	1
4	3	23/01/28	23/01/28	12,3	1	1
5	4	23/01/28	23/01/28	20	2	3

Tabela z danymi TRANSACTIONS:

	ID_TRANSACTION	AMOUNT	AMOUNT_OF_TICKETS	AMOUNT_OF_REDUCED_TICKETS	ID_SEANCE	ID_CUSTOMER
1	3	73,8	6	0	3	1
2	1	30	2	0	1	1
3	2	20	1	0	2	2

Tabela z danymi TEAKENSEATS:

	ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	7	23/01/28	E		4 R	3	(null)	1
2	8	23/01/28	E		5 R	3	(null)	1
3	9	23/01/28	E		6 R	3	(null)	1
4	10	23/01/28	E		7 R	3	(null)	1
5	11	23/01/28	C		3 T	3	3	1
6	12	23/01/28	C		4 T	3	3	1
7	13	23/01/28	C		5 T	3	3	1
8	14	23/01/28	C		6 T	3	3	1
9	15	23/01/28	C		7 T	3	3	1
10	16	23/01/28	C		8 T	3	3	1
11	1	22/02/01	A		5 R	1	(null)	1
12	3	22/02/01	B		4 T	2	(null)	2
13	4	22/02/01	A		7 T	1	(null)	(null)
14	5	22/02/01	B		2 T	1	(null)	(null)
15	6	22/02/01	B		3 T	1	(null)	(null)

Wpiswanie i wybieranie błędnych danych:

Ilość normalnych biletów	<input type="text" value="2"/>
Ilość ulgowych biletów	<input type="text" value="0"/>
Sprzedaj bilety	
Koszt: 24.6 zł	
Info: za dużo biletów	

Ilość normalnych biletów	<input type="text" value="2"/>
Ilość ulgowych biletów	<input type="text" value="0"/>
Sprzedaj bilety	
Koszt: 24.6 zł	
Info: za mało biletów	

Ilość normalnych biletów	<input type="text" value="2"/>
Ilość ulgowych biletów	<input type="text" value="3"/>
Sprzedaj bilety	
Koszt: 54.12 zł	
Info: za dużo biletów	

Wpiswanie i wybieranie poprawnych danych:

Wybranie miejsc:

[Powrót](#)

Ilość normalnych biletów

Ilość ulgowych biletów

[Sprzedj bilety](#)

Koszt: 56,58 zł

Info:
Id klienta: 5

Ekran

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10
H 1	H 2	H 3	H 4	H 5	H 6	H 7	H 8	H 9	H 10

Kupno:

[Powrót](#)

Ilość normalnych biletów

Ilość ulgowych biletów

[Sprzedj bilety](#)

Koszt: 0 zł

Info:

Ekran

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10
H 1	H 2	H 3	H 4	H 5	H 6	H 7	H 8	H 9	H 10

Tabel TRANSACTIONS po wszystkich interakcjach:

ID_TRANSACTION	AMOUNT	AMOUNT_OF_TICKETS	AMOUNT_OF_REDUCED_TICKETS	ID_SEANCE	ID_CUSTOMER
1	3 73,8	6		0	3 1
2	5 56,58	3		2	3 (null)
3	1 30	2		0	1 1
4	2 20	1		0	2 2

Tabel TEAKENSEATS po wszystkich interakcjach:

	ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	30	23/01/28	E		6 T	3	5	(null)
2	31	23/01/28	E		7 T	3	5	(null)
3	27	23/01/28	E		3 T	3	5	(null)
4	11	23/01/28	C		3 T	3	3	1
5	12	23/01/28	C		4 T	3	3	1
6	13	23/01/28	C		5 T	3	3	1
7	14	23/01/28	C		6 T	3	3	1
8	15	23/01/28	C		7 T	3	3	1
9	16	23/01/28	C		8 T	3	3	1
10	28	23/01/28	E		4 T	3	5	(null)
11	29	23/01/28	E		5 T	3	5	(null)
12	1	22/02/01	A		5 R	1	(null)	1
13	3	22/02/01	B		4 T	2	(null)	2
14	4	22/02/01	A		7 T	1	(null)	(null)
15	5	22/02/01	B		2 T	1	(null)	(null)
16	6	22/02/01	B		3 T	1	(null)	(null)

11.8. Dodanie pracownika przez kierownika (EmployeeAddEmployeeController, EmployeeAddEmployeeView.fxml)

Tabela z danymi EMPLOYEES:

	ID_EMPLOYEE	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	HIRE_DATE	FIRE_DATE	ID_POSITION
1	5	Kierownik	Kierownik	Michał Jank...	603446881	23/01/27	(null)		2
2	1	123456	123456	Julixa Noxwako	65734903	22/02/01	(null)		2
3	3	Kasia5	Kasia12345	Pawel Towak	923401903	22/02/01	22/02/01		1
4	4	MartaKK	p55aaaa4	Julian Nowak	567401903	22/02/01	(null)		1

Wpiswanie i wybieranie błędnych danych:

Login jest za krótki
* - pola obowiązkowe

Rejestracja pracownika

*Login:	<input type="text"/>
*Hasło:	<input type="password"/>
*Imię:	<input type="text"/>
*Nazwisko:	<input type="text"/>
* Numer tel.:	<input type="text"/>
*Pozycja:	<input type="text"/>

[Powrót do menu](#)

Rejestruj

Hasło jest za krótki

* - pola obowiązkowe

Rejestracja pracownika

*Login:

*Hasło:

*Imię:

*Nazwisko:

* Numer tel.:

*Pozycja:

[Powrót do menu](#) [Rejestruj](#)

Brak imienia

* - pola obowiązkowe

Rejestracja pracownika

*Login:

*Hasło:

*Imię:

*Nazwisko:

* Numer tel.:

*Pozycja:

[Powrót do menu](#) [Rejestruj](#)

Brak nazwiska

* - pola obowiązkowe

Rejestracja pracownika

*Login:

*Hasło:

*Imię:

*Nazwisko:

* Numer tel.:

*Pozycja:

[Powrót do menu](#) [Rejestruj](#)

ŁY NUMER

* - pola obowiązkowe

Rejestracja pracownika

*Login:	Kierownik
*Hasło:	*****
*Imię:	Bartek
*Nazwisko:	Głowacki
* Numer tel.:	
*Pozycja:	<input type="button" value="▼"/>

Brak danej

* - pola obowiązkowe

Rejestracja pracownika

*Login:	Kierownik
*Hasło:	*****
*Imię:	Bartek
*Nazwisko:	Głowacki
* Numer tel.:	522333111
*Pozycja:	<input type="button" value="▼"/>

Login juz istnieje
 * - pola obowiązkowe

Rejestracja pracownika

*Login:	Kierownik
*Hasło:	*****
*Imię:	Bartek
*Nazwisko:	Główacki
* Numer tel.:	522333111
*Pozycja: <input type="text" value="id_position=1, name=pracownik, sala.etc"/>	
<input type="button" value="Powrót do menu"/> <input style="background-color: #007bff; color: white; border-radius: 5px; border: none; padding: 5px; font-weight: bold;" type="button" value="Rejestruj"/>	

Wpiswanie i wybieranie poprawnych danych:

DODANO PRACOWNIKA
 * - pola obowiązkowe

Rejestracja pracownika

*Login:	Kierownik2
*Hasło:	*****
*Imię:	Bartek
*Nazwisko:	Główacki
* Numer tel.:	522333111
*Pozycja: <input type="text" value="id_position=2, name=kierownik, salary.etc"/>	
<input type="button" value="Powrót do menu"/> <input style="background-color: #007bff; color: white; border-radius: 5px; border: none; padding: 5px; font-weight: bold;" type="button" value="Rejestruj"/>	

Tabel EMPLOYEES po wszystkich interakcjach:

1	5	Kierownik	Kierownik	Michał	Jank...	603446881	23/01/27	(null)	2
2	7	Kierownik2	Kierownik	Bartek	Głów...	522333111	23/01/28	(null)	2
3	1	123456	123456	Julixa	Noxwako	65734903	22/02/01	(null)	2
4	3	Kasia5	Kasia12345	Pawel	Towak	923401903	22/02/01	22/02/01	1
5	4	MartaKK	p55aaaaa4	Julian	Nowak	567401903	22/02/01	(null)	1

11.9. Zwolnienie pracownik przez kierownika (EmployeeEditEmployeeController, EmployeeEditEmployeeView.fxml)

Tabela z danymi EMPLOYEES:

1	5	Kierownik	Kierownik	Michał Jank...	603446881	23/01/27 (null)	2
2	7	Kierownik2	Kierownik	Bartek Głów...	522333111	23/01/28 (null)	2
3	1	123456	123456	Julixa Noxwako	65734903	22/02/01 (null)	2
4	3	Kasia5	Kasia12345	Pawel Towak	923401903	22/02/01 22/02/01	1
5	4	MartaKK	p55aaaaa4	Julian Nowak	567401903	22/02/01 (null)	1

Wpiswanie i wybieranie błędnych danych:

ZWOLNIJ PRACOWNIKA

Potwierdź

POTWIERDŹ

Powrót
ZWOLNIJ PRACOWNIKA

Wpiswanie i wybieranie poprawnych danych:

ZWOLNIJ PRACOWNIKA

Potwierdź

ZWOLNIONO

Powrót
ZWOLNIJ PRACOWNIKA

Tabel EMPLOYEES po wszystkich interakcjach:

ID_EMPLOYEE	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	HIRE_DATE	FIRE_DATE	ID_POSITION
1	5	Kierownik	Kierownik	Michał Jank...	603446881	23/01/27 (null)		2
2	7	Kierownik2	Kierownik	Bartek Głów...	522333111	23/01/28 23/01/28		2
3	1	123456	123456	Julixa Noxwako	65734903	22/02/01 (null)		2
4	3	Kasia5	Kasia12345	Pawel Towak	923401903	22/02/01 22/02/01		1
5	4	MartaKK	p55aaaaa4	Julian Nowak	567401903	22/02/01 (null)		1

11.10. Główny ekran klienta (MainCustomerController, MainCustomerView.fxml)

Brak możliwych testów ręcznych

11.11. Kupno biletu (CustomerBuyController, CustomerBuyView.fxml)

Tabela z danymi SEANCES:

ID_SEANCE	START_TIME	END_TIME	TICKET_PRICE	ID_MOVIE	ID_SCREENING_ROOM
1	8 23/01/29	23/01/29	22	3	2
2	1 22/02/01	22/02/01	15,3	1	2
3	2 22/02/01	22/02/01	20,38	2	1
4	3 23/01/28	23/01/28	12,3	1	1
5	4 23/01/28	23/01/28	20	2	3

Tabela z danymi TRANSACTIONS:

	ID_TRANSACTION	AMOUNT	AMOUNT_OF_TICKETS	AMOUNT_OF_REDUCED_TICKETS	ID_SEANCE	ID_CUSTOMER
1		3 73,8	6	0	3	1
2		5 56,58	3	2	3	(null)
3		1 30	2	0	1	1
4		2 20	1	0	2	2

Tabela z danymi TEAKENSEATS:

	ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	30	23/01/28	E		6 T	3	5	(null)
2	31	23/01/28	E		7 T	3	5	(null)
3	27	23/01/28	E		3 T	3	5	(null)
4	11	23/01/28	C		3 T	3	3	1
5	12	23/01/28	C		4 T	3	3	1
6	13	23/01/28	C		5 T	3	3	1
7	14	23/01/28	C		6 T	3	3	1
8	15	23/01/28	C		7 T	3	3	1
9	16	23/01/28	C		8 T	3	3	1
10	28	23/01/28	E		4 T	3	5	(null)
11	29	23/01/28	E		5 T	3	5	(null)
12	1	22/02/01	A		5 R	1	(null)	1
13	3	22/02/01	B		4 T	2	(null)	2
14	4	22/02/01	A		7 T	1	(null)	(null)
15	5	22/02/01	B		2 T	1	(null)	(null)
16	6	22/02/01	B		3 T	1	(null)	(null)

Wpiswanie i wybieranie błędnych danych:

Ilość normalnych biletów

Ilość ulgowych biletów

Koszt: 0 zł
Info:
za mało biletów

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10

Ilość normalnych biletów
2

Ilość ulgowych biletów
3

Zatwierdź

Kup bilety

Koszt przy zakupie: 96.8 zł
Info:
za dużo biletów

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10

Wpiswanie i wybieranie poprawnych danych:

Kupiono 2 bilety:

Ilość normalnych biletów
0

Ilość ulgowych biletów
0

Zatwierdź

Kup bilety

Koszt: 0 zł
Info:

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10

Zarezerwowano 2 bilety

Ilość normalnych biletów

Ilość ulgowych biletów

Zatwierdź

Kup bilety

Koszt: 0 zł

Info:

A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10
B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10
C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10
E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E 10
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10
G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10

Tabel TRANSACTIONS po wszystkich interakcjach:

ID_TRANSACTION	AMOUNT	AMOUNT_OF_TICKETS	AMOUNT_OF_REDUCED_TICKETS	ID_SEANCE	ID_CUSTOMER
1	73,8	6	0	3	1
2	56,58	3	2	3	(null)
3	44	2	0	8	5
4	30	2	0	1	1
5	20	1	0	2	2

Tabel TEAKENSEATS po wszystkich interakcjach:

ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	30 23/01/28	E		6 T	3	5	(null)
2	31 23/01/28	E		7 T	3	5	(null)
3	32 23/01/29	B		3 T	8	6	5
4	27 23/01/28	E		3 T	3	5	(null)
5	11 23/01/28	C		3 T	3	3	1
6	12 23/01/28	C		4 T	3	3	1
7	13 23/01/28	C		5 T	3	3	1
8	14 23/01/28	C		6 T	3	3	1
9	15 23/01/28	C		7 T	3	3	1
10	16 23/01/28	C		8 T	3	3	1
11	28 23/01/28	E		4 T	3	5	(null)
12	29 23/01/28	E		5 T	3	5	(null)
13	33 23/01/29	B		4 T	8	6	5
14	34 23/01/29	E		5 R	8	(null)	5
15	35 23/01/29	E		6 R	8	(null)	5
16	1 22/02/01	A		5 R	1	(null)	1
17	3 22/02/01	B		4 T	2	(null)	2
18	4 22/02/01	A		7 T	1	(null)	(null)
19	5 22/02/01	B		2 T	1	(null)	(null)
20	6 22/02/01	B		3 T	1	(null)	(null)

11.12. Sprawdzanie rezerwacji (CustomerMyTicketController, CustomerMyTicketView.fxml)

Tabela z danymi TEAKENSEATS:

Wpiswanie i wybieranie poprawnych danych:

Przed usunięciem rezerwacji:

KUPIONE BILETY:

Batman NAPISY 3D 15:59-18:59 miejsce: B3
Batman NAPISY 3D 15:59-18:59 miejsce: B4

ZAREZERWOWANE BILETY

Batman NAPISY 3D 15:59-18:59 miejsce: E5
Batman NAPISY 3D 15:59-18:59 miejsce: E6

[USUŃ REZERWACJE](#)

[Powrót](#)

Po usunięciu:

KUPIONE BILETY:

Batman NAPISY 3D 15:59-18:59 miejsce: B3
Batman NAPISY 3D 15:59-18:59 miejsce: B4

ZAREZERWOWANE BILETY **Usunięto rezerwację**

Batman NAPISY 3D 15:59-18:59 miejsce: E6

[USUŃ REZERWACJE](#)

[Powrót](#)

Tabel TEAKENSEATS po wszystkich interakcjach:

ID_SEAT	EXPIRATION_DATE	ROW_IDENTIFIER	COLUMN_IDENTIFIER	RESERVED_OR_TEAKEN	ID_SEANCE	ID_TRANSACTION	ID_CUSTOMER
1	30 23/01/28	E		6 T		3	5 (null)
2	31 23/01/28	E		7 T		3	5 (null)
3	32 23/01/29	B		3 T		8	6 5
4	27 23/01/28	E		3 T		3	5 (null)
5	11 23/01/28	C		3 T		3	3 1
6	12 23/01/28	C		4 T		3	3 1
7	13 23/01/28	C		5 T		3	3 1
8	14 23/01/28	C		6 T		3	3 1
9	15 23/01/28	C		7 T		3	3 1
10	16 23/01/28	C		8 T		3	3 1
11	28 23/01/28	E		4 T		3	5 (null)
12	29 23/01/28	E		5 T		3	5 (null)
13	33 23/01/29	B		4 T		8	6 5
14	35 23/01/29	E		6 R		8	(null) 5
15	1 22/02/01	A		5 R		1	(null) 1
16	3 22/02/01	B		4 T		2	(null) 2
17	4 22/02/01	A		7 T		1	(null) (null)
18	5 22/02/01	B		2 T		1	(null) (null)
19	6 22/02/01	B		3 T		1	(null) (null)

11.13. Edycja danych klienta(CustomerEditCustomerController, CustomerEditCustomerView.fxml)

Tabela z danymi CUSTOMERS:

ID_CUST...	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1 Natalia	Natalia	Jan	Nowak	758345903	1
2	2 Login66	Haslo66	Pawel	Nowak	958301903	0
3	3 Matryca	Matryca	Juliusz	Slowacki	888561903	1
4	5 Klient	Klient	Paweł	Trybek	223456111	1
5	9 karta33	karta33	Maja	Pawlikow	222333444	1

Wpiswanie i wybieranie błędnych danych:

Edytuj konto

Wpisz w pola dane, które chcesz zmienić:

Hasło:

Imię:

Nazwisko:

Numer telefonu: x

Numer zawiera niedozwolone znaki

Usuń swoje konto

Potwierdź

Potwierdzam

Wpiswanie i wybieranie poprawnych danych:

Edytuj konto

Wpisz w pola dane, które chcesz zmienić:

Hasło:

Imię:

Nazwisko: TATATA

Numer telefonu:

Zmieniono

Obecne dane:

Imię: Paweł

Nazwisko: TATATA

Numer telefonu: 223456111

Następnie usunięto konto a program przeniósł klienta na ekran startowy

Tabel CUSTOMERS po wszystkich interakcjach:

	ID_CUST...	LOGIN	PASSWORD	NAME	SURNAME	PHONE_NUMBER	IS_ACTIVE
1	1	Natalia	Natalia	Jan	Nowak	758345903	1
2	2	Login66	Haslo66	Pawel	Nowak	958301903	0
3	3	Matryca	Matryca	Juliusz	Slowacki	888561903	1
4	5	Klient	Klient	Paweł	TATATA	223456111	0
5	9	karta33	karta33	Maja	Pawlikow	222333444	1

12. Dodatki

Link do GitHuba z kodem aplikacji bazodanowej oraz skryptu sql:

<https://github.com/2MP2/cinema>

13. Spis diagramów i rysunków

Rysunek 1 - Diagram przypadków użycia	6
Rysunek 2 - Poglądowy ERD	14
Rysunek 3 - Konceptualny ERD	17
Rysunek 4 - Fizyczny ERD	23
Rysunek 5. Okno główne aplikacji	37
Rysunek 6. Okno Podglądu przykładowego seansu.	37
Rysunek 7. Okno logowania	38
Rysunek 8. Wybór logowania	38
Rysunek 9. Przykładowy komunikat błędu dla okna logowania	38
Rysunek 10. Okno rejestracji	38
Rysunek 11. Przykładowy komunikat błędu dla okna rejestracji	39
Rysunek 12. Okno główne pracownika	39
Rysunek 13. Okno dodania filmu	40
Rysunek 14. Przykładowy komunikat błędu dla okna dodania seansu	40
Rysunek 15. Okno dodania filmu	40
Rysunek 16. Przykładowy komunikat błędu dla okna dodania filmu	40
Rysunek 17. Okno sprzedaży biletów	41
Rysunek 18. Informacja o id klienta	41
Rysunek 19. Przykładowy komunikat błędu dla okna sprzedaży biletów	41
Rysunek 20. Okno edycji pracowników	42
Rysunek 21. Okno dodawania pracowników	42
Rysunek 22. Przykładowy komunikat błędu dla okna dodania pracownika	42
Rysunek 23. Okno zwolnienia pracownika	43
Rysunek 24. Przykładowy komunikat błędu dla okna zwolnienia pracownika	43
Rysunek 25. Okno główne klienta	43
Rysunek 26. Okno kupna/rezerwacji biletów	44
Rysunek 27. Przykładowy komunikat błędu dla okna kupna/rezerwacji biletów	44
Rysunek 28. Okno rezerwacje klienta	44
Rysunek 29. Okno edycji konta klienta	45
Rysunek 30. Przykładowy komunikat błędu dla zmiany danych klienta	45
Rysunek 31. Przykładowy komunikat błędu dla usunięcia konta klienta	45
Rysunek 32. PU Sprawdzenie listy seansów	46
Rysunek 33. PU Sprawdzenie obłożenia Sali	46
Rysunek 34. PU Kupno biletu	46
Rysunek 35. PU Wybór seansu oraz dostępnego miejsca	47
Rysunek 36. PU Rezerwacja biletu	47

<i>Rysunek 37. PU Odstąpienie od rezerwacji</i>	47
<i>Rysunek 38. PU Sprawdzenia katalogu rezerwacji</i>	48
<i>Rysunek 39. PU Logowanie do systemu</i>	48
<i>Rysunek 40. PU Sprzedaż biletów</i>	48
<i>Rysunek 41. PU Tworzenie seansów</i>	49
<i>Rysunek 42. PU Sprawdzenie dostępności Sali</i>	49
<i>Rysunek 43. PU Dodawanie filmów</i>	49
<i>Rysunek 44. PU Zatrudnianie pracowników</i>	49
<i>Rysunek 45. PU Zwalnianie pracowników</i>	50
<i>Rysunek 46. Cały diagram klas</i>	51
<i>Rysunek 47. Diagram klas dla paczek DTO i Model</i>	52
<i>Rysunek 48. Diagram klas dla DatabaseConnection i Model</i>	53
<i>Rysunek 49. Diagram klas dla wspólnych kontrolerów</i>	54
<i>Rysunek 50. Diagram klas dla kontrolerów pracownika</i>	55
<i>Rysunek 51. Diagram klas dla kontrolerów klienta</i>	56