Multipleks kinowy

Dokumentacja

Wersja 1.1

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:

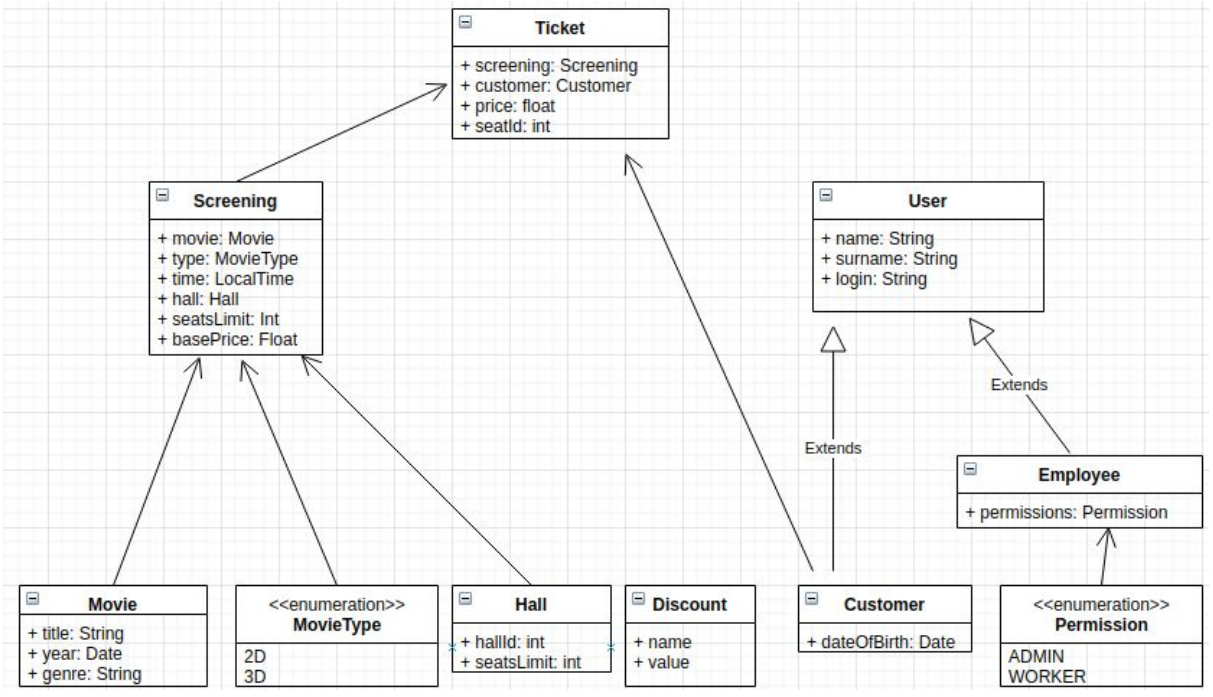
- kompletny model,
- wprowadzanie w aplikacji osób i sal kinowych

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 getery, setery 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 znajdą 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:

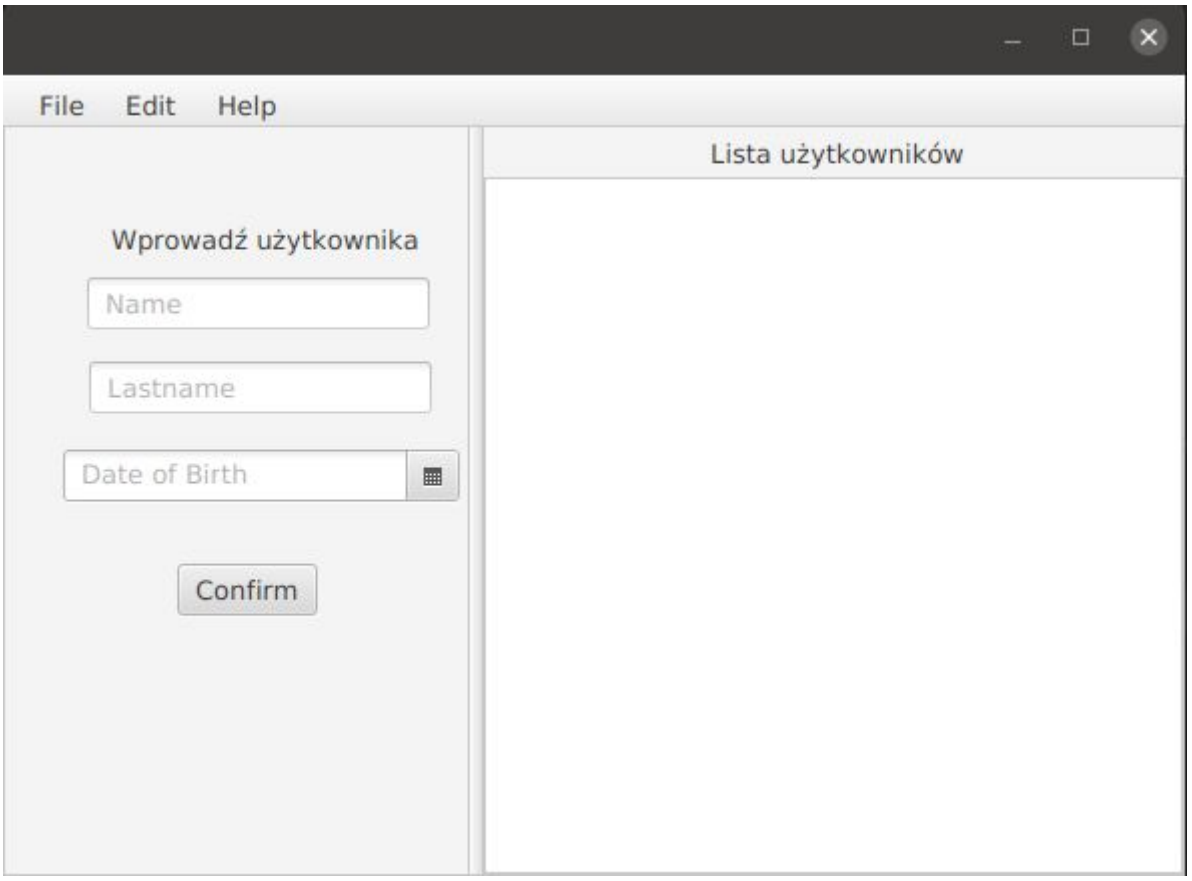
- Movie - Klasa reprezentująca film (rok powstania, tytuł oraz gatunek).
- MovieType - Enum reprezentujący typ filmu. Na chwilę obecną może to być film 2D lub 3D.
- Hall - Sala kinowa. Przechowuje informacje o limicie miejsc.
- Discount - Wszelkiego rodzaju promocje i upusty (wartość może być procentowa 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
2. Podział obowiązków
3. Co zostało zrobione

M3

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