

Projekt ZPO

Movies Database

Michał Mazur

Czw. 15.15 – 17.00

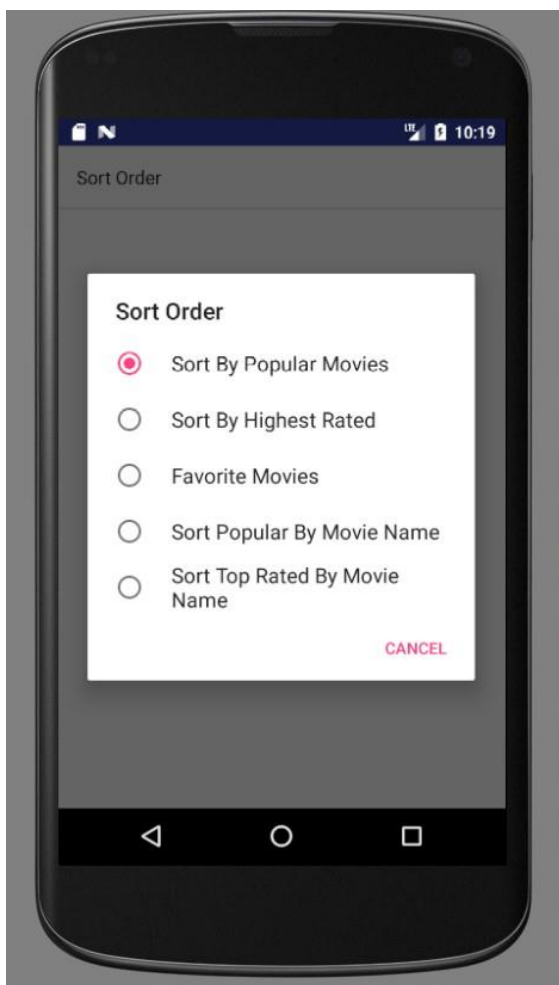
1. Prezentacja działania programu



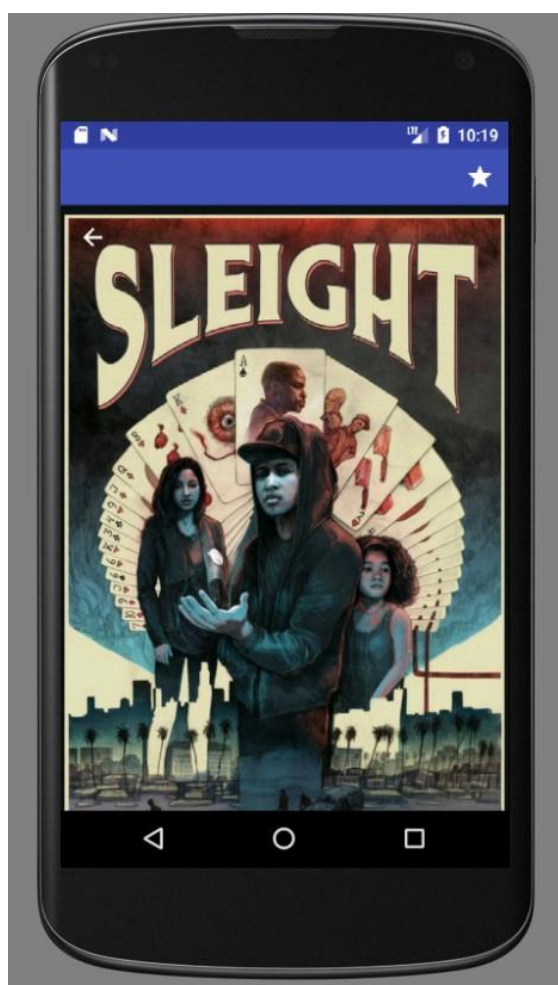
Ekran główny aplikacji



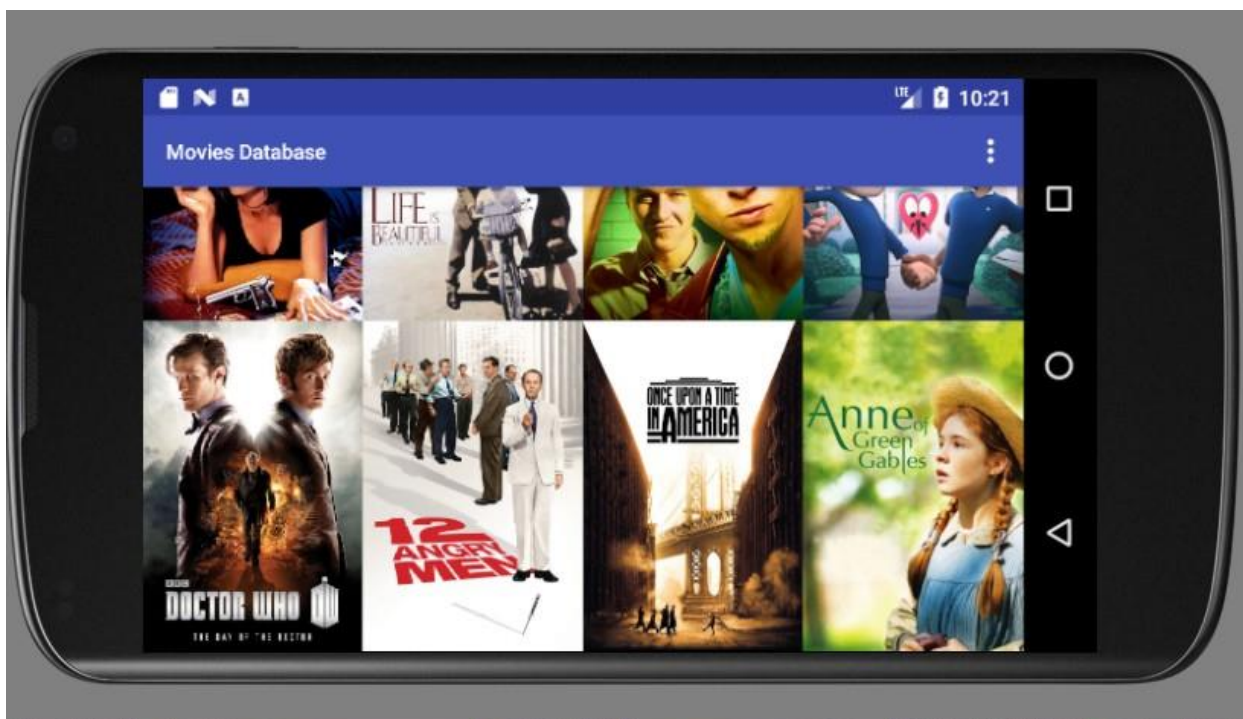
Ekran po kliknięciu na wybrany film



Ekran wyboru sortowania



Ekran prezentujący przycisk „gwiazdkę”



Ekran główny po zmianie orientacji telefonu



Ekran prezentujący szczegóły filmu po zmianie orientacji telefonu

2. Opis działania programu

Program składa się z 6 klas: **FavoriteContract**, **FavoriteDbHelper**, **MainActivity**, **Movie**, **MovieDetailActivity** i **SettingsActivity** oraz 1 interfejsu: **MoviesApiService**. Poniżej przedstawię pokrótce funkcjonalność i działanie każdej z nich:

- 1) **Movie** – klasa zawierająca wszystkie pola i metody o filmie, takie jak np. tytuł, plakat, opis fabuły itd.
- 2) **MovieDetailActivity** – klasa zawierająca wszystkie szczegóły o filmie. Jest używana podczas kliknięcia w dowolny plakat na ekranie głównym – przechodzimy wtedy do opisu filmu oraz mamy możliwość dodać go do „ulubionych”. Pobiera dane z intencji, którą tworzymy w klasie głównej programu – **MainActivity**. Na Toolbarze zawiera przycisk „gwiazdkę”, gdzie po kliknięciu go dany film dodajemy do lokalnej bazy danych – do ulubionych filmów.
- 3) **SettingsActivity** – klasa używana do przechodzenia do opcji programu, w których to z kolei znajduje się wybór sortowania – wyświetlania filmów
- 4) **FavoriteContract** – klasa implementująca interfejs **BaseColumns** (zawierający m. in. kolumnę `_ID`) oraz będąca niejako „podstawką” do klasy **FavoriteDbHelper** – zawieraj podstawowe informacje o bazie danych **SQLite**, takie jak: nazwa tabeli, kolumna z tytułami, kolumna z ścieżkami plakatów oraz kolumna z fabułą filmów.
- 5) **FavoriteDbHelper** – klasa tworząca bazę danych **SQLite**. Zawiera wiele metod służących do dodawania rekordów (`addFavorite`) lub ich usuwania (`deleteFavorite`) oraz innych, takie jak np. metoda `getAllFavorite` służąca do pobrania wszystkich ulubionych filmów (znajdujących się w lokalnej bazie danych).
- 6) **MoviesApiService** – interfejs, który zawiera 2 metody do pobierania API ze strony **TMDb.org**. Jedna umożliwia pobranie najpopularniejszych na ten moment filmów, a druga najlepiej ocenianych.
- 7) **MainActivity** – główna klasa programu, łącząca wszystkie w całość. Zawiera metody takie jak np. tworzenie **REST Adaptera** – umożliwiającego pobranie API, inicjalizację **Retrofita**, który konwertuje API w „czytelną” postać i umożliwia ustawienie listy pobranych filmów

na obiekt klasy MovieAdapter. Ponadto zawiera metodę sprawdzającą orientację telefonu i w zależności od niej ustawia 2 lub 4 kolumny filmów (brak pustego miejsca). Zawiera również metodę sprawdzającą wybraną opcję sortowania – wyświetlania filmów i w zależności od tego wyboru inicjuje różnego typu wyświetlanie (np. popularnych lub ulubionych filmów). Dodatkowo zawiera lokalną klasę MoviesAdapter, która służy do obsługi głównego widoku programu oraz wysyła poprzez mechanizm intencji dane o filmie, które następnie są pobierane w klasie MovieDetailActivity.

Opis typu zwracającego przez metodę **compare** w Comparatorze:

Postać metody:

```
int compare(T o1, T o2)
```

Metoda zwraca wartość typu int – jest to jedna z trzech wartości: -1, 0 lub 1. Jeśli pierwszy argument metody: o1, jest mniejszy niż drugi, to zwraca -1, jeśli jest równy, to zwraca 0, a jeśli jest większy to zwraca 1.

Sam Comparator jest interfejsem, który umożliwia w łatwy sposób sortowanie m. in. obiektów (np. w moim przypadku umożliwia sortowanie alfabetyczne filmów na podstawie ich tytułów).