Obrázok, na ktorom je logo

Automaticky generovaný popis

**Semestrálna práca z predmetu Vývoj aplikácií pre mobilné zariadenia**

Vypracoval: Juraj Juríček

Skupina: 5ZYS31

# Popis a analýza riešeného problému

Aplikácia je navrhnutá s cieľom poskytnúť užívateľom prostriedky na správu a organizáciu ich filmových zoznamov. S rastúcim počtom filmov dostupných na rôznych platformách a v rôznych žánroch je dôležité mať nástroj, ktorý pomôže užívateľom sledovať a spravovať filmy, ktoré ich zaujímajú. Moja aplikácia ponúka užívateľom možnosť pridávať filmy do zoznamov, filtrovať ich podľa rôznych kritérií a udržiavať históriu sledovaných filmov.

## Špecifikácia zadania:

* Cieľom aplikácie je umožniť užívateľom vytvárať a spravovať zoznamy filmov.
* Užívatelia môžu pridávať filmy do svojich zoznamov a uvádzať informácie o filmoch, ako je názov, režisér, obsadenie, žáner a ďalšie.
* Aplikácia poskytuje možnosť filtrovania filmov podľa rôznych kritérií, ako je žáner a názov.
* Užívatelia môžu pridávať filmy do svojej histórie, aby si zapamätali filmy, ktoré už videli.
* Používateľské rozhranie je prehľadné a jednoduché na používanie.

## Podobné aplikácie

**IMDb (Internet Movie Database)**

Obrázok, na ktorom je text, snímka obrazovky, multimediálny softvér, softvér

Automaticky generovaný popis IMDb je veľmi populárna platforma, ktorá poskytuje informácie o filmoch, hercoch, režiséroch a ďalších súvisiacich údajoch. Umožňuje užívateľom vytvárať vlastné zoznamy filmov, hodnotiť filmy a získavať aktuálne informácie o filmovom priemysle. Moja aplikácia sa od IMDb líši tým, že je zameraná viac na správu a organizáciu filmových zoznamov pre jedného použivateľa, a poskytuje jednoduché a prehľadné prostredie pre sledovanie a filtrovanie filmov.

**Moviebase**

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popisMoviebase je ďalšia populárna aplikácia, ktorá slúži na sledovanie filmov a správu zoznamov. Užívateľom umožňuje pridávať filmy, sledovať ich, priraďovať hodnotenia a udržiavať prehľad o svojich obľúbených filmoch.

**Just Watch**

Obrázok, na ktorom je text, snímka obrazovky, multimediálny softvér, softvér

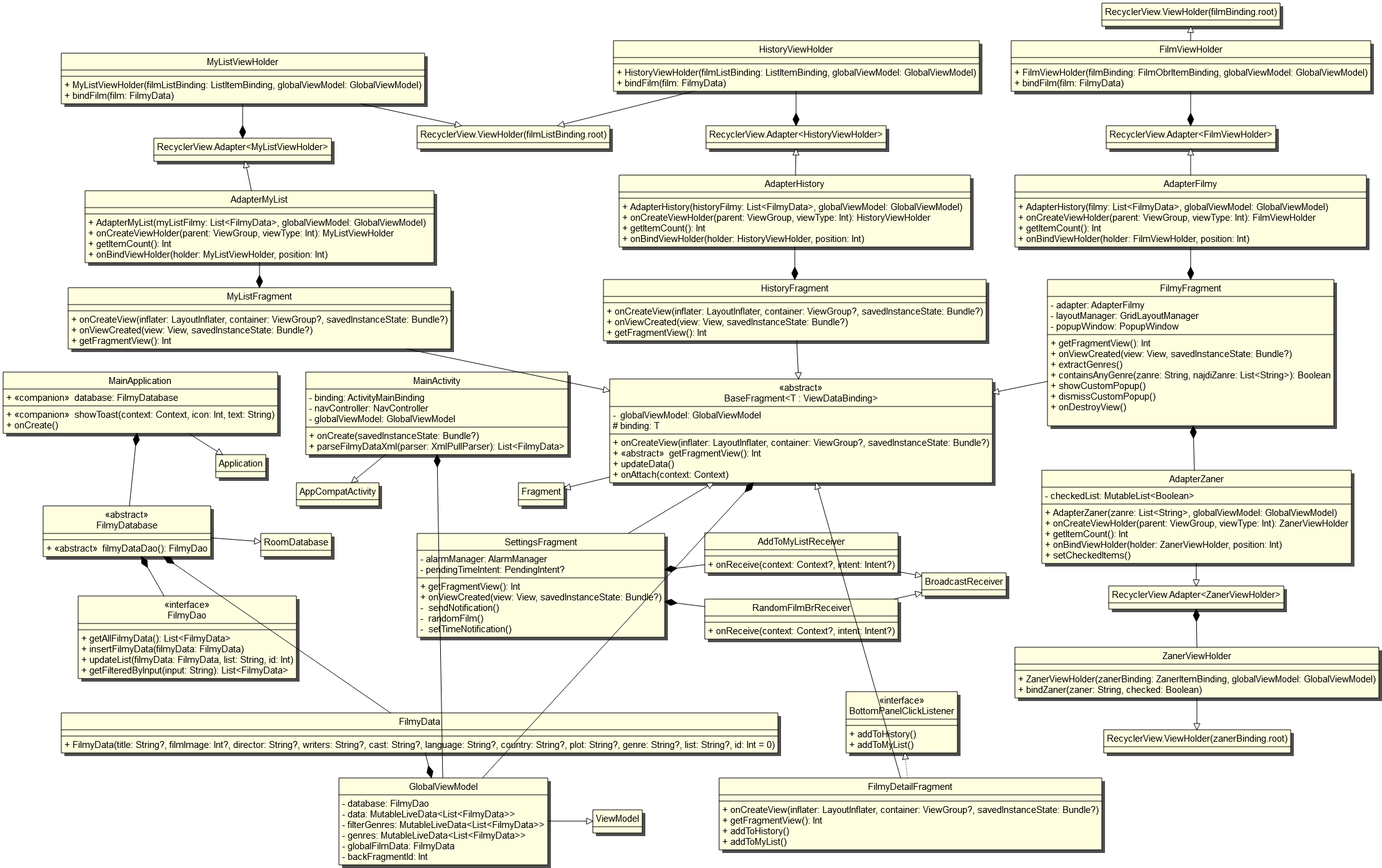
Automaticky generovaný popisJust Watch je populárna aplikácia, ktorá poskytuje informácie o dostupnosti filmov a televíznych relácií na rôznych platformách. Užívateľom umožňuje vyhľadávať filmy a relácie, zobraziť, na ktorých streamingových službách sú dostupné, a poskytuje informácie o ich cenách a ďalších podrobnostiach.

# Návrh riešenia problému

**Krátka analýza**

Obrázok, na ktorom je text, diagram, rad, písmo

Automaticky generovaný popisPo spustení aplikácie sa používateľovi zobrazí hlavná obrazovka v ktorej môže vyhľadávať a filtrovať filmy podľa žánru a taktiež si môže zobraziť informácie o týchto filmoch a následne si ich môže pridať do svojich listov. Ďalej na spodnom panely má prístup k panelu kde sa môže prekliknúť na prehliadky svojich listov a do nastavenia kde si môže nastaviť notifikácie.



# Popis implementácie

Obrázok, na ktorom je text, koláž, ľudská tvár, grafický dizajn

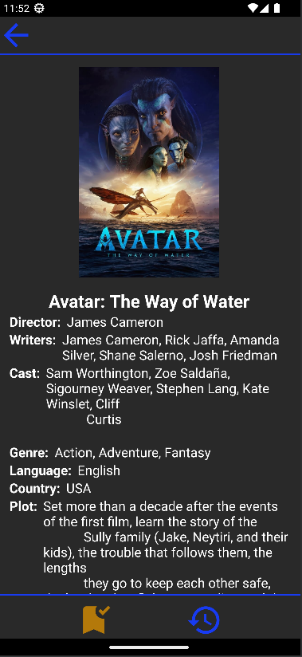
Automaticky generovaný popisObrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis**Fragment filmov**

V rámci implementácie fragmentu Filmov som využil technológiu RecyclerView na zobrazenie dvoch resp štyroch radov kariet vedľa seba podľa toho či je obrazovka nastavená. Vytvoril som dynamický zoznam, ktorý automaticky prispôsobuje svoju veľkosť a obsah podľa dostupných údajov.

Ďalej je tu implementované vyhľadávanie pomocou SearchView a žánrov pomocou Popup Menu. Používateľ môže zadávať hľadané výrazy a v reálnom čase sa zobrazujú výsledky, ktoré sú relevantné pre zadaný výraz.

Okrem toho som vytvoril Popup Menu pre výber filmov podľa žánru. Používateľ má možnosť vybrať a zaškrtnúť žánre, podľa ktorých chce vyhľadávať z Popup menu a následne sa zobrazia iba filmy, ktoré patria do vybraných žánrov.

 **Fragment detailov filmu**

V rámci implementácie tohto fragmentu som vytvoril zobrazenie informácií o filme. Tieto informácie zahŕňajú názov filmu, režiséra, zápletka, žánre a iné relevantné údaje.

Okrem zobrazenia informácií som implementoval dve tlačidlá, ktoré umožňujú používateľovi pridať film do dvoch rôznych zoznamov a to zoznam filmov, ktoré si chce pozrieť, a zoznam už pozretých filmov. Tieto tlačidlá sú navrhnuté tak, aby boli interaktívne a reagujú na používateľove kliknutia.

V rámci implementácie som vytvoril funkcionalitu pri pridávaní filmov do zoznamov ktorá mení farbu tlačidla, ak je film už prítomný v niektorom zo zoznamov. Toto vizuálne odlíšenie pomáha používateľovi rozpoznať, či je film už pridaný do zoznamu.

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

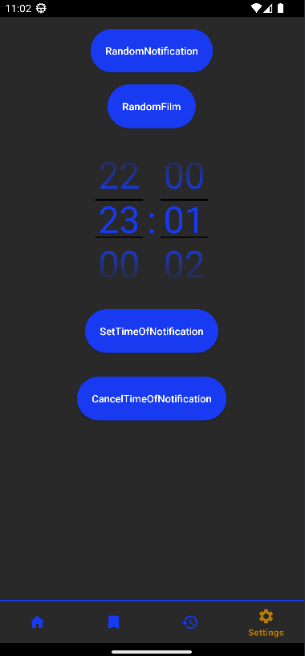
Automaticky generovaný popisOkrem toho som pridal aj oznámenia v podobe vlastne nadizajnovaného Toastu, ktorý sa zobrazuje po kliknutí na tlačidlá. Tento Toast informuje používateľa o tom, čo sa práve deje so filmom. Napríklad, ak používateľ pridá film do zoznamu na pozretie, Toast zobraziť správu "Film pridaný do zoznamu na pozretie". Podobne, ak používateľ označí film ako už pozretý, Toast zobrazí správu "Film pridaný do histórie".

**Fragmenty zoznamov**

V rámci implementácie som vytvoril fragmenty pre zobrazenie zoznamov filmov, ktoré obsahujú RecyclerView pre prezeranie filmov v daných zoznamoch.

Pre každý zoznam som vytvoril samostatný fragment, ktorý obsahuje RecyclerView ako hlavný zobrazovací prvok. RecyclerView spolupracuje s adaptérom, ktorý spravuje dáta a zabezpečuje správne vykresľovanie jednotlivých položiek.

**Fragment nastavení**

V rámci implementácie fragmentu som vytvorili tlačidlá, ktoré umožňujú používateľovi nastaviť notifikáciu s náhodným filmom na základe zvoleného času k čomu bol použitý BroadcastReceiver. Po stlačení tlačidla pre nastavenie notifikácie sa zobrazí Toast s informáciou o úspešnom nastavení notifikácie.

Okrem toho som tiež implementoval tlačidlo na zrušenie tejto notifikácie, ak by už bola nastavená.

Ďalej som implementoval tlačidlo pre okamžité zobrazenie notifikácie, čo umožňuje používateľovi ihneď vidieť notifikáciu s náhodne vybraným filmom.

Notifikácia s náhodným filmom obsahuje názov filmu a jeho zápletku. Mimo to obsahuje aj funkčné prvky ako tlačidlo, ktoré slúži na pridávanie filmu do zoznamu "Filmov na pozretie" k čomu bol použitý druhý BroadcastReceiver a po stlačení notifikácie sa nám otvorí aplikácia.

V rámci fragmentu som tiež umiestnil tlačidlo, ktoré umožňuje zobraziť informácie o filme. Po jeho stlačení sa zobrazí podrobný popis filmu, vrátane jeho názvu, zápletky a ďalších relevantných informácií. Toto tlačidlo umožňuje používateľovi získať viac informácií o filme a môže rozhodnúť, či ho chce pridať do zoznamu alebo nie.

Tlačidlá a vyberač času boli navrhnuté tak, aby sa zhodovali so štýlom aplikácie a zapadali do jej celkového vizuálneho dizajnu.

**BaseFragment**

Trieda BaseFragment je abstraktná trieda, ktorá slúži ako základná implementácia pre všetky fragmenty v aplikácii. Poskytuje spoločnú funkcionalitu a metódy, ktoré sú použité vo všetkých fragmentoch.

Medzi hlavné vlastnosti a metódy triedy BaseFragment je to že inicializuje binding fragmentu ktorý túto triedu dedí čo uľahčuje a zjednodušuje prácu s fragmentami. Ďalej obsahuje inštanciu triedy globalViewModel, ktorá sa vytvára po pripojení fragmentu k aktivite.

Abstraktná metóda getFragmentView vracia ID layoutu fragmentu. Toto ID layoutu sa používa pri inflatovaní layoutu v metóde onCreateView.

Metóda updateData slúži na aktualizáciu údajov vo view modely. Asynchrónne vykonáva aktualizáciu zoznamu údajov v samostatnej korutine.

**GlobalViewModel**

Trieda GlobalViewModel je trieda, ktorá slúži ako globálny view model v aplikácii. Je vytvorená na zdieľanie dát medzi rôznymi fragmentmi aplikácie. Trieda dedí ViewModel a umožňuje uchovávať a spravovať dáta.

Hlavným účelom GlobalViewModel je uchovávať dáta o filmoch a žánroch, ktoré sú potrebné pre rôzne časti aplikácie. Tieto dáta sú reprezentované pomocou MutableLiveData, ktoré umožňuje sledovať zmeny v dátach a aktualizovať ich v reálnom čase.

**GlobalViewModel obsahuje hodnotky ako:**

* **data** je MutableLiveData obsahujúce zoznam filmových dát. Tieto dáta môžu byť aktualizované a sledované fragmentmi, ktoré ich využívajú.
* **filterGenres** je MutableLiveData obsahujúce žánre filmov, podľa ktorých sa vykonáva filtrovanie. Tento zoznam môže byť aktualizovaný, čo spôsobuje zmeny vo filtrovaní filmov.
* **genres** je MutableLiveData obsahujúce zoznam všetkých žánrov filmov. Tieto dáta sa používajú pre výber žánrov pri filtrovaní filmov.
* **globalFilmData** je premenná, ktorá uchováva aktuálne vybrané údaje o filme. Tieto údaje sú použité v rôznych častiach aplikácie, napríklad pri zobrazovaní podrobností o filme.
* **backFragmentId** je premenná, ktorá uchováva identifikátor fragmentu, na ktorý sa má vrátiť po stlačení tlačidla späť.
* **database**, je inštanciou triedy FilmyDao. Táto premenná slúži na prístup k lokálnej databáze filmových údajov.

**Room databáza**

Trieda FilmyDatabase reprezentuje databázu pre filmy. Je to implementácia rozhrania RoomDatabase z knižnice Room, ktorá poskytuje možnosti práce s relačnou databázou.

V databáze je použitá entita FilmyData, ktorá slúži ako dátová trieda obsahujúca kompletné informácie o filme. Táto entita je kľúčovým prvkom pre ukladanie, získavanie, aktualizovanie a filtrovanie filmov v rámci aplikácie.

FilmyData zoskupuje všetky atribúty týkajúce sa filmu vrátane názvu, obrázka, režiséra, scenáristov, obsadenia, jazyka, krajiny pôvodu, zápletky a žánrov. Okrem toho, táto entita tiež uchováva informáciu o príslušnom zozname, v ktorom sa film nachádza, čo môže byť napríklad označenie ako "history" alebo "myList". Identifikátor filmu je automaticky generovaný a slúži na jednoznačné identifikovanie záznamu v databáze.

Všetky vlastnosti (title, filmImage, director, writers, cast, language, country, plot, genre, list, id) sú anotované pomocou @ColumnInfo, čo definuje ich názvy v databáze. Vlastnosť id je označená ako primárny kľúč s automatickým generovaním hodnoty (@PrimaryKey(autoGenerate = true)).

Rozhranie FilmyDao zabezpečuje operácie, ako je získanie všetkých filmových údajov z databázy (getAllFilmyData()), vloženie nových filmových údajov do databázy (insertFilmyData()), aktualizácia existujúcich filmových údajov v databáze (updateData()), aktualizácia zoznamu v ktorom sa film nachádza (updateList()), a získanie filtrovaných filmových údajov na základe zadaného vstupu (getFilteredByInput()).

**MainApplication**

MainApplication je hlavnou triedou aplikácie a rozširuje triedu Application. Slúži na inicializáciu niektorých dôležitých komponentov aplikácie a poskytuje prístup k nim cez statické metódy a premenné z iných častí kódu.

V rámci tejto triedy je implementovaná metóda showToast, ktorá umožňuje zobrazenie Toast správy s definovanými parametrami. Metóda showToast prijíma kontext aplikácie, identifikátor ikonky a textový obsah správy. Pomocou LayoutInflater sa vytvorí vlastný layout pre Toast správu, kde sa ikona a text nastavia podľa zadaných parametrov. Potom je Toast správa zobrazená pomocou toast.show().

Trieda MainApplication obsahuje aj statickú premennú database, ktorá je inštanciou databázy pre filmy (FilmyDatabase). Táto premenná je inicializovaná v metóde onCreate, ktorá sa volá pri spustení aplikácie. Inicializácia databázy prebieha pomocou metódy databaseBuilder, ktorá vytvára inštanciu databázy na základe definovaného typu (FilmyDatabase) a názvu databázy ("MyFilmDatabase.db"). Následne je táto inštancia priradená do statickej premennej database, aby sa mohla použiť v rôznych častiach aplikácie.

**MainActivity**

je hlavnou aktivitou aplikácie. V tejto triede sa nachádza metóda onCreate, ktorá sa volá pri vytvorení aktivity. V tejto metóde sa vykonáva inicializácia a konfigurácia rôznych komponentov. Ako je navController, ktorý slúži na nastavenie navigácie medzi fragmentmi pomocou spodného panelu s tlačidlami v aktivite a je konfigurovaný s activity\_main\_nav\_host\_fragment pre zobrazenie fragmentu.

Ďalej sa tu nachádza vytvorenie notifikačného kanála pre posielanie notifikácií. A nakoniec je tu metóda parseFilmyDataXml, ktorá spracúva XML dáta pomocou XmlPullParser a vytvára zoznam objektov typu FilmyData. Táto metóda prechádza XML dokument, číta jednotlivé tagy a ukladá hodnoty do príslušných premenných. Keď narazí na ukončovací tag pre FilmyData, vytvorí objekt FilmyData s hodnotami zo spracovaného XML a pridá ho do zoznamu, ktorý nakoniec vráti.

# Zdroje

Popup menu: <https://www.youtube.com/watch?v=R4o5VEs-iKI>

Toast: <https://www.youtube.com/watch?v=ayjINChcG3g>

Notifikácia: <https://www.youtube.com/watch?v=mWb_hEBLIqA>

ViewModel, Livedata: <https://www.youtube.com/watch?v=N7J27pBTtuI>

Coroutine: <https://www.youtube.com/watch?v=VWlwkqmTLHc>

ROOM: <https://youtu.be/bOd3wO0uFr8>

SearchView: <https://youtu.be/M73Vec1oieM>