Aplicatie MANAGEMENT FILME

Prof. Coordonator:

Alexandru DITA

Realizat de:

Alexandru-Mihai Codreanu

Gr. 1076

Contents

[Introducere 2](#_Toc29126571)

[Definire problema 2](#_Toc29126572)

[Definire obiective 2](#_Toc29126573)

[Definire cadru general 2](#_Toc29126574)

[Descrierea solutiei 2](#_Toc29126575)

[Descriere scenariu de utilizare 2](#_Toc29126576)

[Descriere functionalitati 2](#_Toc29126577)

[Implementare solutie 3](#_Toc29126578)

[Avantare si probleme intalnite la implementarea aplicatiei folosind Android 5](#_Toc29126579)

[Bibliografie 6](#_Toc29126580)

# Introducere

## Definire problema

Problema pe care aplicatia de Management Filme doreste sa o resolve este aceea de a ajuta un consummator de filme in a-si organiza filmele nou aparute si a avea o lista clara cu ce acesta doreste sa vada.

## Definire obiective

Proiectul, pentru a rezolva problema, prezinta utilizatorului o lista de filme ce pot fi adaugate la favorite spre a fi salvate local, pentru ca acesta sa poata sa salveze timp atunci cand doreste sa vizioneze un film. Totodata, utilizatorul poate sa filtreze filmele salvate in functie de ratingul minim pe care il doreste, satisfacand dorinta de a vedea un film de calitate.

## Definire cadru general

Mediul in care aplicatia se doreste a fi folosita este in cel familial, atunci cand utilizatorul doreste sa vizioneze un film de care s-a interesat in prealabil si la momentul respective nu a avut timp sa il vizioneze.

# Descrierea solutiei

## Descriere scenariu de utilizare

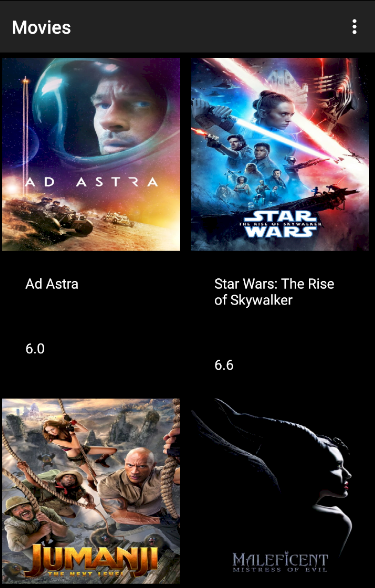
Modul in care aplicatia a fost destinate sa fie utilizata este urmatorul: un utilizator care simte nevoia de organizarea filmelor pe care le doreste sa le vada isi creeaza un cont ca apoi sa adauge in lista de favorite filmele care il intereseaza sau care par a fi pe gustul lui dupa ce a vazut posterul, ratingul de pe IMDB sau a citit descrierea filmului. Utilizatorul revine la un momentdat, cand doreste sa vada un film pentru a selecta filmul pe placul lui din cele salvate in cadrul aplicatiei, limitand riscul ca acesta sa uite filmul care i-a starnit odata interesul.

## Descriere functionalitati

Functionalitatile pe care le prezinta aplicatia sunt: inregistrarea/logarea in cont pentru a salva filmele, prezentarea sumara a filmelor nou aparute (Titlu, poster si rating IMDB), prezentarea amanuntita a filmelor (Titlu, poster, rating IMDB si descrierea filmului), adaugarea filmelor in lista de favorite, filtrarea filmelor salvate in functie de rating, recomandarea unui cinematograf, schimbarea parolei si stergerea contului si prezentarea unei statistici bazata pe filme si de cate ori a fost adaugat unul in lista de favorite.

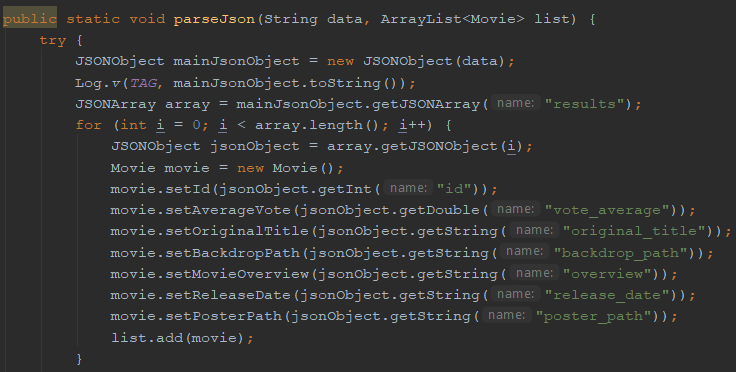
# Implementare solutie

Aplicatia a fost dezvoltata folosind Android Studio si a consumat API-ul pentru filme pus la dispozitie de The Movie DB.

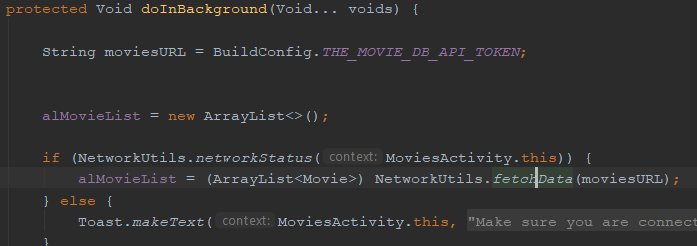
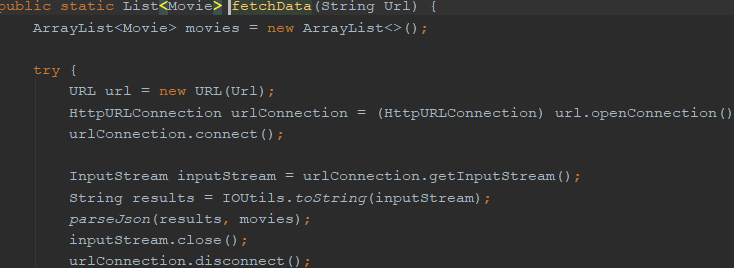
 API-ul The Movie DB pune la dispozitie un JSON care cuprinde informatii despre filme, din care am ales sa folosesc ID-ul, titlul, descrierea, ratingul, data lansarii si posterul filmului.

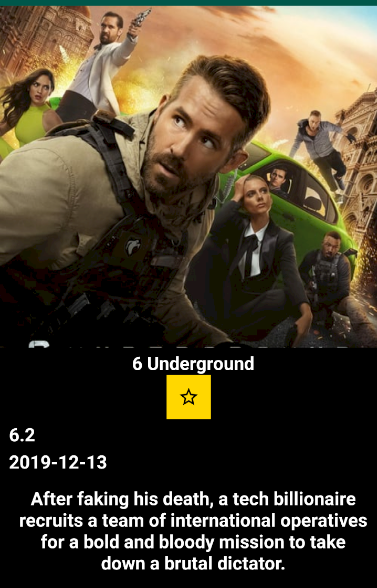
Posterul filmului il preiau din JSON sub forma unui string de tipul “xBHvZcjRiWyobQ9kxBhO6B2dtRI.jpg” pe care il concatenez la un link standard oferit de TMDB (“https://image.tmdb.org/t/p/w500”) care, accesat in cadrul proiectului returneaza posterul.

Inregistrarea, logarea si adaugarea filmelor in lista de favorite se fac local, folosind SQLite, aplicatia avand doua tabele: “user” ( ce contine ID-ul care este PK, username-ul si parola utilizatorului) si “favorite” ( ce contine ID-ul inregistrarii care este PK, username-ul utilizatorului ce a adaugat in favorite filmul ca FK, ID-ul filmului, titlul, ratingul, posterul si descrierea filmului).

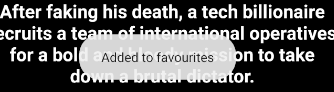
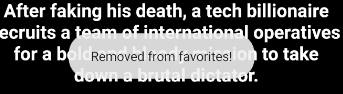
 Totodata, fiecare user inregistrat este preluat si de Firebase pentru ca apoi sa fie afisati toti userii in cadrul activitatii *LiveUsersActivity*.

Dupa inregistrare/logare utilizatorului ii este prezentata activitatea *(MoviesActivity)* ce contine lista de filme nou aparute. Activitatea contine un Swipe Refresh Layout pe care este aplicat un RecyclerView care, pe orizontal, contine doua Card View-uri in care sunt incarcate datele sumare ale filmului: Posterul, Titlul si Ratingul.

Datele sunt preluate din JSON folosind metoda *GetMoviesAsync,* o metoda asincrona care se foloseste de *ParseJSON* pentru a prelua informatiile din API si a le trimite mai departe in *MoviesActivity*.

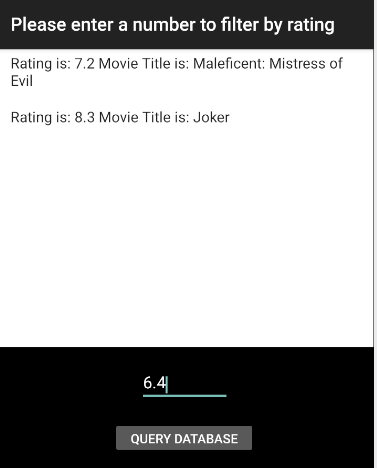
 Odata gasit un film, se deschide activitatea specifica filmului care, pe langa titlu, poster si rating contine si prezentarea filmului. Informatiile din cadrul activitatii detaliate sunt primite prin Intent din Adapter-ul clasei pentru filme.

Daca se doreste salvarea filmului ca favorit, dupa apasarea butonului se face o verificare in baza de date care returneaza daca filmul este deja in favorite sau nu si afiseaza mesajul corespunzator.



In cadrul Option Menu-lui din dreapta sus din *MoviesActivity*  exista trei optiuni, anume de *SettingsActivity, RateAppActivity* si *About Activity.* RateAppActivity introduce SharedPrefferences pentru a salva rating-ul sub forma de stelute in cadrul fisierului de preferinte,

*SettingsActivity* prezinta controale variate in cadrul aplicatiei:

***FILTER YOUR MOVIES***este o optiune care ofera posibilitatea utilizatorului sa isi filtreze filmele selectate dupa rating.

In cadrul EditText-ului care accepta doar caractere numerice se introduce un numar care va fi folosit apoi in Query-ul pe baza de date in functie de *username-ul* utilizatorului logat si *rating-ul* pe care acesta il introduce: *db.rawQuery("select \* from favorite where username =? and rating >= ?", new String[] {username.toLowerCase(),value});*

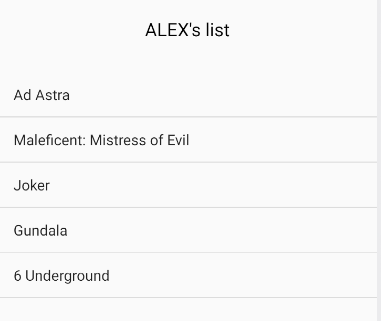
***CINEMA RECOMMANDATION*** este o optiune care foloseste API-ul Google Maps pentru a oferi recomandarea unui cinematograf, anume cel din AFI Palace Cotroceni.

***SHOW GRAPH*** este o optiune care arata utilizatorului de cate ori a fost adaugat un film de catre ceilalti utilizatori, prin intermediul bazei de date locale:

*Cursor cursor = db.rawQuery("SELECT movieid, COUNT(\*) as nrids, title " +*

*"FROM favorite " +*

*"GROUP BY movieid",new String [] {});*

***ACCOUNT SETTINGS*** este o optiune care ii permite utilizatorului fie sa isi schimbe parola contului, fie sa isi stearga contul.

***SHOW FAVORITES*** este o optiune care ii prezinta utilizatorului raportul cu filmele pe care le-a adaugat la favorite, intr-o activitate *MovieUserActivity* ce contine un ListView in care sunt incarcate filmele din baza de date in urma query-ului: *Cursor cursor = db.rawQuery("select \* from favorite where username= ?", new String[] {username.toLowerCase()});*

***SHOW LIVE USERS*** este o optiune care prezinta utilizatorului toti utilizatorii care s-au inregistrat in aplicatie. Rezultatul este returnat din Firebase si introdus intr-un ListView.

***DISCONNECT*** duce utilizatorul inapoi in prima pagina a aplicatiei.

## Avantare si probleme intalnite la implementarea aplicatiei folosind Android

Avantajul principal pe care l-am recunoscut in urma implementarii aplicatiei in Android a fost multitudinea de librarii care au facilitate si usurat implementarea unor componente ale aplicatiei: RecycleView, CardView, ButterKnife, Picasso.

Problema principala cu care m-am intalnit de-a lungul dezvoltarii proiectului a fost modul in care functioneaza Debugger-ul, mai exact multitudinea de clase si metode care se apeleaza in background. Am descoperit, totusi, ca Debugger-ul poate fi evitat prin folosirea Toast si a Log-urilor.

Opinii

Consider ca, la momentul actual, o aplicatie de genul, pentru sistemul de operare Android, utilizarea acestei solutii (Android Studio) este cea ideala, avand in vedere lipsa experientei in acest domeniu.

Totodata, Android Studio ofera un pachet intuitiv, care este gata de utilizare si pentru care se gasesc resurse si raspunsuri rapid.

# Bibliografie

<https://stackoverflow.com/questions/26562848/how-to-integrate-google-maps-in-android-app>;

<https://medium.com/@onyekweretrust/how-to-create-a-simple-graph-in-android-6c484324a4c1>;

<https://firebase.google.com/docs/database/android/start>;