Movie Tracker Application

# Scopul aplicației

Scopul aplicatiei Movie Tracker este de a oferi utilizatorilor un mod convenient de a gestiona experienta lor de vizionare a filmelor. Acest lucru include facilitarea urmaririi filmelor vizionate si accesul la informatii despre filme si cinematografe din apropiere. Aplicatia ofera o interfata intuitiva si usor de utilizat pentru a realiza acest scop.

Aplicatia Movie Tracker poate fi folosita de orice persoana interesata de gestionarea experientei de vizionare a filmelor. Acest lucru poate include iubitori de filme, cinefili sau orice persoana care doreste sa tina evidenta filmelor vizionate si sa aiba acces la informatii despre filme si cinematografe. Aplicatia poate fi utilizata pe dispozitive Android.

# API

Pentru partea de back-end a aplicației, s-au folosit urmatoarele:

* ASP.NET Core (.NET 6, C# 10)
* Entity Framework
* BD SQL Server
* Servicii cloud în Azure
  + Azure SQL Database
  + Azure App service

Alegerea tehnologiilor utilizate pentru creearea API-ului folosit de aplicatia Android este motivata pe de-o parte de performata framework-ului .NET, care suporta cele mai multe request-uri in comparatie cu alte tehnologii similare – vezi pagina [asta](https://dusted.codes/how-fast-is-really-aspnet-core) pentru mai multe detalii, precum si de rapiditatea dezvoltarii proiectului, pe de alta parte.

# Baza de date

Baza de date a aplicatiei este creata cu ajutorul Entity Framework Core in maniera code-first, plecand de la clasele model (Movie, Cinema, etc.) si este stocata in cloud, folosind Azure SQL Database. Astfel baza de date este comuna pentru toti utilizatorii, inca din faza de development a aplicatiei, neexistand discrepante intre datele pe care programatorii le folosesc.

Azure ofera posibilitatea unui query editor in browser, insa este totusi recomandata utilizarea SQL Server Management Studio (SMMS) local pentru a beneficia de o functionalitate completa.

Avanatajul utilizarii Azure SQL Database este acela ca, fiind un serviciu de tip Platform as a Service (PaaS), nu este nevoie sa ne preocupam de mentanata serverului de baze de date, operatiile precum actualizari sau instalarea de patch-uri fiind realizata de catre Microsoft.

Query-urile sunt realizate in C#, folosind LINQ, in interiorul unor metode din clase repository.

Pentru utilizarea Enitity Framework este de asemenea necesara instalarea pachetelor NuGet:

* Microsoft.EntityFrameworkCore
* Microsoft.EntityFrameworkCore.SqlServer
* Microsoft.EntityFrameworkCore.Tools

String-ul de conexiune la baza de date este stocat in fisierul userSecrets.json pentru a se asigura faptul ca datele de conectare la BD nu vor fi commit-uite, fisierul de secrete existand local, pentru fiecare utilizator, la alta cale (path) decat cea a proiectului.

# Endpoint-uri

Proiectul API expune endpoint-urile care accepta request-urile de la aplicatia client prin intermediul celor trei controllere:

* CinemasController.cs
* MoviesController.cs
* UsersController.cs

Majoritatea endpoint-urilor din aceste controllere sunt de tipul GET si POST, continand metode de tipul find (dupa id-ul unei resurse) si de actualizare de date (marcarea unui film ca vizionat de catre un utilizator).

Trebuie mentionat ca in proiectul API exista si metode care nu sunt utilizate de catre aplicatia Android.

Lista tuturor endpoint-urilor disponibile, precum si testarea functionalitatii acesteia se poate realiza prin accesarea paginii: <https://localhost:7062/swagger/index.html> in cazul in care ASP porneste pe portul 7062.

Pentru utilizarea swagger este necesare pachetul NuGet Swashbuckle.AspNetCore precum si adaugarea urmatoarei configurari in clasa Program.cs:

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

# Hosting-ul API-ului

Pentru a putea utiliza proiectul API fara sa fie nevoie sa rulam proiectul local, am ales sa publicam proiectul in Azure app service.

Acesta este un PaaS care permite managementul aplicatiilor web, permitand mentinearea si actualizarea de configurari, aspecte de networking, securitate sau Application Insights (notificari pentru probleme legate de aplicatie, logging). Ofera de asemenea o rata a disponibilitatii (eng. availability) de 99.99%.

Un alt motiv pentru utilizarea acestui serviciu este pentru evitarea avertismentelor de securitate de catre Volley in aplicatia Android, deoarece ruland api-ul local nu avem un certificat de securitate pentru localhost.

# Frontend-ul aplicației

Aplicatia Movie Tracker este realizata folosind Android Studio si este dezvoltata in limbajul de programare Java. Acest lucru permite integrarea cu platforma Android si accesul la un set larg de biblioteci si unelte pentru dezvoltarea de aplicatii mobile. Java este un limbaj de programare puternic si versatil, cu o comunitate activa de dezvoltatori, ceea ce il face potrivit pentru a dezvolta aplicatii complexe, cum ar fi aplicatia Movie Tracker.

Aplicația este compusă din următoarele:

1. **StartActivity:**

* Afișează o animație în momentul pornirii aplicației.

1. **MainActivity:**

* Conține 4 fragmente: HomeFragment, WatchedListFragment, MoviesFragment, și CinemasFragment

1. **HomeFragment:**

* Afiseaza pagina principala a aplicatiei.

1. **WatchedListFragment:**

* Afișează o listă de filme pe care utilizatorul le-a vizionat

1. **MoviesFragment:**

* Permite utilizatorului să răsfoiască filme

1. **CinemasFragment:**

* Afișează o listă de cinematografe din apropierea locației utilizatorului

1. **Navigation:**

* Fragmentele pot fi navigate folosind bottom navigation bar

1. **Functionality:**

* Aplicația își propune să gestioneze experiența de vizionare a filmelor a unui utilizator, inclusiv urmărirea filmelor vizionate și găsirea de informații despre filme și cinematografele din apropiere.

# Tehnologii folosite

Aplicatia Movie Tracker foloseste aceste tehnologii: Java, Android Studio, Picasso, si Volley. Java este limbajul de programare utilizat pentru a dezvolta aplicatia, in timp ce Android Studio este mediu de dezvoltare integrat (IDE) utilizat pentru a crea aplicatii Android. Picasso si Volley sunt biblioteci open-source care ajuta la incarcarea si afisarea imaginiilor si la gestionarea cererilor HTTP, respectiv. Acestea sunt tehnologii cheie care contribuie la crearea unei aplicatii eficiente si cu o experiență utilizatorilor optima.

1. **Java**: Java este un limbaj de programare puternic, orientat pe obiecte, care este utilizat larg in dezvoltarea aplicatiilor. Java ofera flexibilitate si scalabilitate, permitand dezvoltatorilor sa construiasca aplicatii complexe cu un cod usor de intretinut.
2. **Android Studio**: Android Studio este un mediu de dezvoltare integrat (IDE) pentru platforma Android, creat de Google. Android Studio ofera o serie de instrumente si caracteristici care ajuta la crearea aplicatiilor Android, cum ar fi suportul pentru debug, instrumente de editare si editare visuala, si acces la biblioteci de dezvoltare Android.
3. **Picasso**: Picasso este o biblioteca open-source pentru Android care permite incarcarea si afisarea de imagini eficient. Picasso se ocupa de toate aspectele incarcarii imaginilor, precum cache-ul si optimizarea, astfel incat dezvoltatorii sa se poata concentra pe functionalitatile principale ale aplicatiei.
4. **Volley**: Volley este o biblioteca de rețea dezvoltată de către Google care ofera o metoda simpla si eficienta de a gestiona cereri HTTP si a prelucra raspunsurile. Volley permite dezvoltatorilor sa execute cereri HTTP asincrone, astfel incat aplicatia sa poata raspunde rapid la interacțiunile utilizatorilor.

Acestea sunt tehnologii esentiale care pot fi utilizate pentru a dezvolta aplicatia Movie Tracker si a oferi o experiență utilizatorilor optima. Utilizarea acestor tehnologii poate ajuta la crearea unei aplicatii rapide, eficiente si cu o interfata intuitiva.

In Android, o activitate este o componenta de baza care reprezinta o ecran complet al aplicatiei. O activitate poate afisa informatii si poate gestiona interactiunea utilizatorului prin intermediul elementelor grafice cum ar fi butoane, campuri de text si liste.

Un fragment, pe de alta parte, este o componenta de interfata utilizator care poate fi utilizata in cadrul unei activitati. Un fragment poate fi gandit ca o componenta modulara care poate fi utilizata pentru a imparti o activitate in mai multe sectiuni si pentru a oferi functionalitati specifice. De exemplu, un fragment poate afisa o lista de produse sau poate oferi optiuni de filtrare pentru produsele afisate.

In aplicatia Movie Tracker, fragmentele sunt utilizate pentru a oferi utilizatorilor diferite functionalitati cum ar fi gestionarea listei de filme vizionate sau afisarea informatiilor despre cinematografele din zona. Prin intermediul acestor fragmente, aplicatia poate fi impartita in mai multe sectiuni si poate oferi utilizatorilor o experienta mai usor de utilizat.

# Funtionalitatea aplicației

1. **StartActivity**: Aceasta activitate contine o animatie care poate fi afisata la pornirea aplicatiei. Scopul acestei activitati este de a oferi utilizatorilor o experienta placuta in timp ce aplicatia se incarca si se pregateste pentru utilizare.
2. **MainActivity**: Aceasta activitate este cea mai importanta activitate a aplicatiei si contine mai multe fragmente: HomeFragment, WatchedListFragment, MoviesFragment si CinemasFragment. Fiecare fragment ofera functionalitati specifice si poate fi afisat in cadrul activitatii MainActivity.

* **HomeFragment**: Acest fragment poate fi utilizat pentru a afisa informatii generale despre aplicatie, cum ar fi lista de filme recomandate sau informatii despre ultimele lansari de filme. La selectarea unui film din acest top, va fi apelata activitatea MovieManagerActivity, care va oferi informatii specifice filmului. Acest fragment contine CardViews.

Un **CardView** este o componenta grafica din Android care poate fi utilizata pentru a afisa informatii intr-un format similar unui card. Un CardView poate fi gandit ca o cutie care poate contine elemente precum imagini, text si butoane, toate organizate intr-un mod atractiv si usor de utilizat.

CardView-urile sunt folosite frecvent in aplicatiile Android pentru a afisa informatii intr-un format elegant si intuitiv, iar prin intermediul acestora, utilizatorii pot interactiona cu informatiile afisate prin intermediul elementelor grafice continute de carduri.

In aplicatia Movie Tracker, CardView-urile este utilizat pentru a afisa informatii despre filmele din top, cum ar fi imaginea, titlul si recenziile utilizatorilor.

* **WatchedListFragment**: Acest fragment poate fi utilizat pentru a afisa o lista de filme vizionate de utilizatori. Pentru a afisa aceste filme, aplicatia foloseste un CustomAdapter.

Un **CustomAdapter** este o clasa Java personalizata care poate fi utilizata pentru a afisa datele intr-un format personalizat intr-o lista sau un grid. CustomAdapter-ul poate fi utilizat pentru a personaliza aspectul si comportamentul fiecarui element din lista sau grid, precum si pentru a defini interactiunile utilizatorului cu acestea.

In cazul aplicatiei Movie Tracker, myAdapter este utilizat pentru a afisa filmele vizionate prin intermediul unei liste. Prin intermediul acestui adapter personalizat, aplicatia poate defini cum arata fiecare element din lista de filme vizionate, cum ar fi imaginea si titlul filmului.

* **MoviesFragment**: Acest fragment poate fi utilizat pentru a afisa o lista de filme disponibile pentru vizionare. In acest scop, aplicatia foloseste un GridView, care afiseaza informatiile in format de grila. **ImageAdapter** este utilizat pentru a seta imaginile afisate in GridView. Acesta poate fi un adapter personalizat care poate fi utilizat pentru a afisa imaginile si alte informatii specifice filmelor in GridView.
* **CinemasFragment**: Acest fragment poate fi utilizat pentru a afisa informatii despre cinematografele din zona utilizatorului, precum si programul de filme si informatii despre bilete.

Aceste activitati si fragmente sunt componentele cheie ale aplicatiei Movie Tracker si ofera functionalitatile necesare pentru a gestiona filmele si a oferi utilizatorilor o experienta placuta.

# Viitoare funcționalități

Aplicația Movie Tracker are posibilitatea de a fi extinsă. Aceata poate contine urmatoarele funtionalitati:

1. **Cautare avansata**: o functie de cautare care sa permita utilizatorilor sa filtreze filmele in functie de gen, data de lansare sau rating.
2. **Recenzii utilizatorilor**: o sectiune care sa permita utilizatorilor sa scrie si sa citeasca recenzii despre filmele vizionate.
3. **Reminder**: o functie care sa permita utilizatorilor sa seteze un reminder pentru a fi notificati cand un film dorit este disponibil in cinematografe sau pe platforme online.
4. **Integrare cu platforme de streaming**: posibilitatea de a verifica disponibilitatea filmelor pe platforme de streaming populare, precum Netflix.

Acestea sunt doar cateva exemple de functionalitati care ar putea fi adaugate in viitor la aplicatia Movie Tracker. Dezvoltarea ulterioara a aplicatiei poate fi adaptata in functie de cerintele utilizatorilor si de tendintele din domeniu.