

---

# SMB116-Projet d'application Android

Date de rendu : 17/05/2025

## VUE D'ENSEMBLE

Dans le cadre de ce projet, vous allez concevoir et développer une application Android permettant à un utilisateur de gérer sa liste de films préférés. L'application interagit avec l'API **The Movie Database (TMDB)** pour récupérer des informations sur les films, afficher des films tendance, permettre à l'utilisateur de rechercher des films, et gérer une liste de visionnage personnalisée. L'application inclura également un système d'authentification local et un profil personnalisé avec des informations comme la photo de profil et les genres de films favoris.

## OBJECTIFS

Le projet a plusieurs objectifs pédagogiques :

1. **Conception et développement d'une application Android complète :**

- Apprendre à utiliser des composants fondamentaux d'Android, comme les activités, la navigation, les interactions avec une API, et le stockage local avec Room etc.

2. **Développer des compétences professionnelles :**

- Mettre en place une architecture logicielle maintenable.
- Utiliser des bonnes pratiques de développement Android, y compris la gestion de code source, la qualité du code, et la gestion des performances.

---

## CARACTÉRISTIQUES

L'application doit disposer des fonctionnalités suivantes :

### Inscription et Connexion :

- Lors de la première utilisation de l'application, l'utilisateur doit s'inscrire en renseignant un email, un mot de passe, ainsi que sa date de naissance à l'aide d'un composant graphique adapté.
- Les informations de l'utilisateur seront enregistrées localement à l'aide de **Room** pour la gestion des utilisateurs.
- Lors de l'ouverture suivante de l'application, l'utilisateur pourra se connecter en saisissant son email et son mot de passe. L'application validera ces informations et l'utilisateur pourra accéder à l'application avec une session persistée (les informations de connexion étant stockées localement).

### Paramétrage du profil utilisateur :

- Après la première connexion, l'utilisateur sera dirigé vers une page de paramétrage de son profil où il pourra :
  - **Prendre une photo de profil** : L'utilisateur pourra utiliser la caméra de l'appareil pour capturer une photo ou bien choisir une photo existante dans sa galerie. La photo sera ensuite sauvegardée localement en tant que fichier au sein de l'application.
  - **Sélectionner ses genres de films favoris** : L'utilisateur pourra choisir ses genres de films préférés (ex : action, comédie, science-fiction, etc.). Ces informations seront enregistrées dans **Room** et pourront être utilisées pour personnaliser l'expérience de l'utilisateur (par exemple, pour afficher des films correspondant à ses goûts).
- À tout moment dans l'application, l'utilisateur pourra revenir sur cet écran de paramétrage pour modifier ses informations de profil (photo et genres favoris).

---

## Écran des tendances :

- L'application devra afficher une liste des films tendances actuels. Cette liste sera récupérée à partir de l'API TMDB et affichera des informations sommaires pour chaque élément. Ces informations incluent :
  - **Image de l'affiche**
  - **Nom du film**
  - **La date de sortie ou de diffusion**
  - **Les principaux genres associés au film**
  - **Sa popularité** (si disponible)
- Chaque film affiché pourra être sélectionné, permettant à l'utilisateur de consulter plus de détails sur un écran spécifique à ce film (par exemple, synopsis complet, acteurs, etc.).

## Recherche d'un film :

L'application permettra à l'utilisateur de rechercher des films de deux façons :

### 1. Recherche par nom :

L'utilisateur pourra entrer le nom d'un film dans un champ de recherche. Les résultats correspondants seront affichés sous forme de liste, avec des informations sommaires sur chaque film (image de l'affiche, titre, etc.). L'utilisateur pourra alors sélectionner un film pour en voir plus de détails.

### 2. Découverte par genres favoris :

En fonction des genres de films enregistrés dans le profil de l'utilisateur (via la page de paramétrage), l'application propose une liste de films associés à ces genres. Cette fonctionnalité permet à l'utilisateur de découvrir de nouveaux films en fonction de ses préférences, avec un affichage des résultats similaire à la recherche par nom.

## Liste de films à visionner :

L'application permettra à l'utilisateur d'ajouter des films à sa **WatchList**, une liste personnelle de films qu'il souhaite visionner plus tard. Cette fonctionnalité sera accessible à tout moment et depuis n'importe quel écran où un film est présenté.

- **Ajouter un film à la WatchList :**

L'utilisateur pourra ajouter un film à sa WatchList à partir de l'écran de détails d'un film. Un bouton ou une action permettra d'ajouter/retirer le film à la liste de visionnage de l'utilisateur.

---

- **Consulter la WatchList :**

L'utilisateur pourra consulter sa WatchList depuis un écran dédié. Cet écran affichera tous les films ajoutés à la liste, avec les mêmes informations sommaires que celles présentées dans les écrans de tendances ou de recherche (image de l'affiche, titre, date de sortie, etc.).

- **Gestion de la WatchList :**

L'utilisateur peut ajouter et retirer des films de sa WatchList. Cette gestion sera facile et intuitive.

- **Stockage local :**

La liste sera sauvegardée localement à l'aide de Room, permettant de conserver la liste entre les différentes sessions de l'utilisateur.

### **Partage d'un film :**

- Depuis la page de détail d'un film, l'utilisateur pourra partager un lien vers la page du film sur le site de **TMDB**. Ce lien prendra la forme suivante :

[https://www.themoviedb.org/movie/{id\\_movie}](https://www.themoviedb.org/movie/{id_movie})

L'utilisateur pourra partager ce lien via les mécanismes de partage natifs d'Android (Intents), permettant ainsi de l'envoyer par message, email, ou via des applications de réseaux sociaux. Cette fonctionnalité permet à l'utilisateur de recommander facilement des films à ses amis ou de partager ses découvertes.

### **Rappel pour un film :**

L'application permet à l'utilisateur de programmer un rappel pour un film ajouté à sa WatchList.

- Depuis l'écran de détail d'un film, l'utilisateur pourra appuyer sur un bouton **"Me rappeler"**.
- Une interface lui permettra alors de sélectionner une **date et une heure** à laquelle il souhaite être notifié.
- À l'heure définie, une **notification** lui rappellera de regarder le film.
- La notification affichera le **titre du film** et permettra d'accéder directement à sa fiche dans l'application.

---

## RÉFLEXION ET ORGANISATION DU PROJET

Cette application combine plusieurs aspects du développement Android, notamment :

- **Requêtes API** pour récupérer des films depuis TMDB.
- **Stockage local** avec Room pour sauvegarder les utilisateurs et la WatchList.
- **Gestion des notifications** pour rappeler un film à voir.
- **Navigation** entre les différents écrans de l'application.

Avant de se lancer dans le développement, il est recommandé de **prendre le temps de réfléchir** à une stratégie globale et à l'**architecture logicielle** de l'application. Une bonne organisation du code et des responsabilités de chaque composant facilitera la maintenance et l'évolutivité du projet.

Quelques questions à se poser pour structurer le projet :

- **Architecture générale :**
  - Quel modèle d'architecture adopter ?
  - Comment organiser les différentes couches de l'application (données, UI, logique métier) ?
- **Gestion des données :**
  - Comment structurer les entités et relations dans Room ?
  - Quels types de requêtes API seront nécessaires et comment éviter de surcharger le réseau ?
  - Comment synchroniser les données locales avec l'API TMDB, notamment pour la WatchList
- **Navigation et interface utilisateur :**
  - Comment organiser les écrans et la navigation entre eux ?
  - Quels composants UI utiliser pour une expérience fluide ?
- **Fonctionnalités spécifiques :**
  - Comment programmer et gérer les notifications pour les rappels de films ?

Prendre le temps de répondre à ces questions facilitera grandement la mise en œuvre du projet et aidera à éviter des erreurs architecturales difficiles à corriger par la suite.

---

## Critères de notations

### 1. Fonctionnalités implémentées (8 points)

L'application doit répondre aux exigences définies dans le cahier des charges :

- **Inscription/Connexion avec stockage local** (2 pts)
- **Gestion du profil utilisateur (photo + genres favoris)** (1,5 pts)
- **Écran des tendances et recherche de films (API TMDB)** (2 pts)
- **WatchList avec ajout/suppression de films** (1,5 pts)
- **Système de rappel avec notifications** (1 pt)

**Critère d'évaluation :** Fonctionnalités opérationnelles, absence de bugs bloquants.

### 2. Qualité de l'architecture et du code (5 points)

- **Utilisation d'une architecture propre** (2 pts)
- **Organisation du code et séparation des responsabilités** (1,5 pts)
- **Bonne gestion des appels API et stockage local (éviter surcharge...)** (1,5 pts)

**Critère d'évaluation :** Code lisible, structuré, réutilisable et respectant les principes SOLID et les bonnes pratiques Kotlin.

### 3. Expérience utilisateur et ergonomie (4 points)

- **Interface fluide et bien pensée (Jetpack Compose bien utilisé)** (2 pts)
- **Navigation intuitive et fluide** (1 pt)
- **Bonne gestion des états et feedback utilisateur (chargement, erreurs, etc.)** (1 pt)

**Critère d'évaluation :** Application agréable à utiliser, transitions fluides, pas de comportements frustrants.

### 4. Gestion du projet et bonnes pratiques (3 points)

- **Utilisation de Git et bonne gestion des commits** (1 pt)
- **Code bien commenté et documentation minimale** (0,5 pt)
- **Application testée et relativement stable (pas de crashes évidents)** (1,5 pts)

**Critère d'évaluation :** Démarche professionnelle dans le développement et la gestion du code.

---

## Ressources

L'application utilisera l'API public de The Movie Database (TMDB), il sera nécessaire de créer un compte pour obtenir une clé API. Attention, il est préférable de ne pas mettre cette clé API sur votre système de gestion de source. Vous trouverez ci-dessous les liens utiles :

- [Inscription TMDB](#)
- [Enregistrement clé API](#)
- [Documentation développeur TMDB](#)

Je vous conseille de vous pencher sur des librairies existantes pour gérer certains aspects de votre application, inutile de réinventer la roue. Libre à vous cependant d'utiliser celles de votre choix.

Requêtes API :

- <https://square.github.io/retrofit/>
- <https://square.github.io/okhttp/>

Affichage des images depuis une URL public :

- <https://coil-kt.github.io/coil/>
- <https://github.com/bumptech/glide>

Stockage des données :

- <https://developer.android.com/training/data-storage/shared-preferences>
- <https://developer.android.com/training/data-storage/room>