

TP Flutter Netflim 8

1- Analyse du projet

1.1 Analyse de l'arborescences

Utiliser la commande tree (répertoire lib uniquement) afin de détailler l'architecture des fichiers du projet . → Mettre l'arborescence dans votre compte rendu.

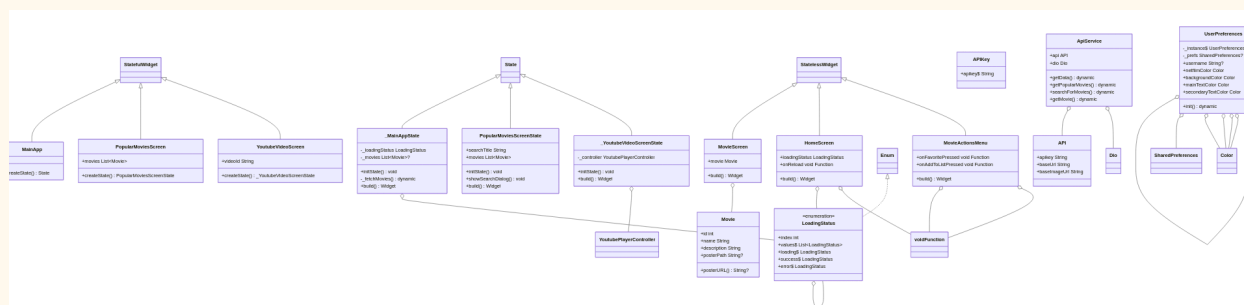
```
lib
├── main.dart
├── models
│   └── movie.dart
├── screens
│   ├── home_screen.dart
│   ├── movie_screen.dart
│   ├── popular_movies_screen.dart
│   └── youtube_video_screen.dart
├── services
│   ├── api.dart
│   ├── api_key.dart
│   ├── api_service.dart
│   └── user_preferences.dart
└── widgets
    └── movie_actions_menu.dart
```

→ Compléter le tableau suivant en répartissant les fichiers et dossiers du projet dans les catégories appropriées du modèle MVC :

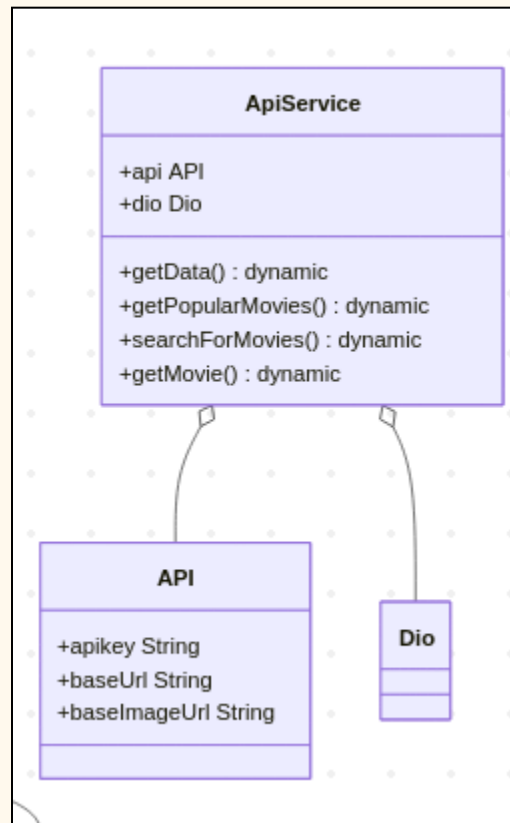
Modèle (Model)	Vue (View)	Contrôleur (Controller)
/models/movie.dart (class Movie structure pour stocker les informations. main.dart Les données manipulées Les fichiers API sont de la manipulations des données	/screens/fichier_screen.dart (Affichage des éléments titre,synopsys etc /widgets/movie_actions_menu.dart affiche des éléments main.dart (widgets)	main.dart _MainAppState agit comme un contrôleur.

1.2 Analyse des classes

1.2.1 Aller à l'url <https://mermaid.live> puis y copier/coller le contenu du fichier texte. → Sauvegarder le diagramme au format image et le mettre dans votre compte rendu.



1.2.2 Analyser le diagramme de classes et l'arborescence du projet. →
Quelle classe et bibliothèque est utilisée pour réaliser les accès http à l'API REST de TMDB.



→ Quelle classe est utilisée pour afficher la première page?

La classe utilisé pour afficher la première page c'est HomeScreen

→ Quelle classe est utilisée pour afficher la liste des films?

La classe utilisé est PopularMoviesScreen et utilisé en tant que List

→ Quelle classe est utilisée pour afficher le détail d'un film?

La classe utilisé est MovieScreen pour les détail

2- Amélioration du projet

2.1 Amélioration de la fiche détails

→ Suivre le tutoriel annexe 3. Puis améliorer le rendu des nouveaux champs - Faire en sorte que cela ait du style. - et réaliser une capture d'écran de la page de détails finalisée. Fournir également son code source.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <!--
5   If you are serving your web app in a path other than the root, change the
6   href value below to reflect the base path you are serving from.
7
8   The path provided below has to start and end with a slash "/" in order for
9   it to work correctly.
10
11   For more details:
12   * https://developer.mozilla.org/en-US/docs/Web/HTML/Element/base
13
14   This is a placeholder for base href that will be replaced by the value of
15   the '--base-href' argument provided to 'flutter build'.
16 -->
17 <base href="/">
18
19 <meta charset="UTF-8">
20 <meta content="IE=Edge" http-equiv="X-UA-Compatible">
21 <meta name="description" content="A new Flutter project.">
22
23 <!-- iOS meta tags & icons -->
24 <meta name="apple-mobile-web-app-capable" content="yes">
25 <meta name="apple-mobile-web-app-status-bar-style" content="black">
26 <meta name="apple-mobile-web-app-title" content="netflim">
27 <link rel="apple-touch-icon" href="icons/Icon-192.png">
28
29 <!-- Favicon -->
30 <link rel="icon" type="image/png" href="favicon.png"/>
31
32 <title>netflim</title>
33 <link rel="manifest" href="manifest.json">
34
35 <script>
36   // The value below is injected by flutter build, do not touch.
37   const serviceWorkerVersion = null;
38 </script>
39 <!-- This script adds the flutter initialization JS code -->
40 <script src="flutter.js" defer></script>
41 </head>
42 <body>
43 <script>
44   window.addEventListener('load', function(ev) {
45     // Download main.dart.js
46     _flutter.loader.loadEntrypoint({
47       serviceWorker: {
48         serviceWorkerVersion: serviceWorkerVersion,
49       },
50       onEntrypointLoaded: function(engineInitializer) {
51         engineInitializer.initializeEngine().then(function(appRunner) {
52           appRunner.runApp();
53         });
54       }
55     });
56   });
57 </script>
58 </body>
59 </html>

```

2.2 Accès au trailer

A l'aide de la documentation officielle du site TheMovieDB, récupérer la bande annonce des films

(<https://developer.themoviedb.org/reference/movie-videos> et exemple en annexe 2) et afficher là en exploitant la classe `YoutubeVideoScreen`.

→ Détailler les actions effectuées et le code modifié

Récupération des vidéos des bandes d'annonce

```
class API {
    /// La clé API pour accéder à l'API de TMDb. Stockée dans un fichier
    final String apikey = APIKey.apikey;

    /// L'URL de base pour les requêtes API vers TMDb.
    final String baseUrl = 'https://api.themoviedb.org/3';

    /// L'URL de base pour accéder aux images sur TMDb, permettant de co
    final String baseImageUrl = 'https://images.tmbd.org/t/p/w500';

    /// L'URL pour les bandes-annonces de films
    final String baseVideoUrl = 'https://www.youtube.com/watch?v=';

    /// L'URL pour récupérer les vidéos d'un film spécifique.
    Future<List<dynamic>> fetchMovieVideos(int movieId) async {
        final url = Uri.parse('$baseUrl/movie/$movieId/videos?api_key=$api
        final response = await http.get(url);

        if (response.statusCode == 200) {
            final data = json.decode(response.body);
            return data['results'];
        } else {
            throw Exception('Failed to load videos');
        }
    }
}

class MovieService {
    final API api = API();

    Future<List<dynamic>> getMovieVideos(int movieId) async {
        return await api.fetchMovieVideos(movieId);
    }
}
```

Modification de la class Movie pour y ajouter videoId

```
class Movie {
  final int id;
  final String name;
  final String description;
  final String? posterPath;
  final String? tagline;
  final String? releaseDate;
  final List<String>? genres;
  final int? runtime;
  final double? voteAverage;
  final String? videoId;

  Movie({
    required this.id,
    required this.name,
    required this.description,
    this.posterPath,
    this.tagline,
    this.releaseDate,
    this.genres,
    this.runtime,
    this.voteAverage,
    this.videoId,
  });
```

modification api_service.dart pour y récupérer la video

```
/// Récupère les vidéos d'un film en fonction de son `movieId`
Future<List<dynamic>> getMovieVideos(int movieId) async {
  try {
    Response response = await dio.get(
      '${api.baseUrl}/movie/$movieId/videos',
      queryParameters: {
        'api_key': api.apikey,
        'language': 'fr-FR',
      },
    );

    if (response.statusCode == 200) {
      return response.data['results'];
    } else {
      throw Exception('Échec de récupération des vidéos');
    }
  } catch (e) {
    rethrow;
  }
}
```

Modification de `movie_screen.dart` pour ajouter la récupération de la bande d'annonce puis le bouton pour y accéder.

```
// FutureBuilder pour afficher les vidéos (bandes-annonces)
Padding(
  padding: const EdgeInsets.all(16.0),
  child: FutureBuilder<List<dynamic>>(
    future: _movieVideos,
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
        return Center(child: CircularProgressIndicator());
      }

      if (snapshot.hasError) {
        return Center(child: Text('Erreur: ${snapshot.error}'));
      }

      if (!snapshot.hasData || snapshot.data!.isEmpty) {
        return Center(child: Text('Aucune vidéo disponible.'));
      }

      // Si des vidéos sont disponibles, on affiche le bouton pour la bande-annonce
      List<dynamic> videos = snapshot.data!;
      var trailer = videos.firstWhere(
        (video) => video['type'] == 'Trailer',
        orElse: () => null,
      );

      return trailer != null
        ? ElevatedButton(
            onPressed: () {
              // Naviguer vers YoutubeVideoScreen avec l'ID de la vidéo
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => YoutubeVideoScreen(
                    videoId: trailer['key'],
                  ), // YoutubeVideoScreen
                ), // MaterialPageRoute
              );
            },
            child: Text('Voir la bande-annonce'),
            style: ButtonStyle(
              backgroundColor: MaterialStateProperty.all(
                UserPreferences().netflimColor,
              ), // ButtonStyle
            ) // ElevatedButton
          ): Center(child: Text('Pas de bande-annonce trouvée.'));
    }
  );
}
```