Application Mobile

***Fiche technique :***

**Présentation de projet et de l’équipe**

L’équipe est composée d’Enzo Marini et Raphaël Martin  
Le projet consistera à créer une application mobile permettant d’avoir accès à une « Bibliothèque » des séries d’animations japonaise et à une documentation.

**Nom** :

AniMania

**Résumé de l’application :**

Cette application recensera toutes les séries animées disponible sur Myanimelist et à ses informations

Il pourra donc naviguer sur l’application et trouver les informations concernant ses séries favorites

**Framework utilisé** :

Android Studios.

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

**Base de données :**

Pour plus de simplicité et de praticité, la base de données sera en local (en cours de développement)

**Langage Utilisé :**

Avec l’utilisation du Framework Android Studio, nous sommes limités dans le choix du langage. Nous utiliserons donc Kotlin. C’est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java, JavaScript, et vers plusieurs plateformes en natif

**API utilisé :**

Pour notre application, nous avons utilisé l’API Jikan qui est une API open-source se basant sur le site communautaire le plus actif d’animation japonaise et manga Myanimelist

Code pour l’intégration de l’API :

interface AnimeService {  
  
 @GET("top/anime")  
 fun getTopAnimes(): Call<TopAnime>  
  
 @GET("search/anime")  
 fun getSearchedAnime(@Query("q")queryString: String): Call<SearchedAnime>  
  
 companion object {  
 val BASE\_URL = "https://api.jikan.moe/v3/"  
  
 fun create(): AnimeService {  
 val retrofit = Retrofit.Builder()  
 .addConverterFactory(GsonConverterFactory.create())  
 .baseUrl(BASE\_URL)  
 .build()  
 return retrofit.create(AnimeService::class.*java*)  
 }  
 }  
  
}

Pour l’application, nous utilisons la dépendance Retrofit2

* Retrofit transforme votre API REST en une interface Java. Il utilise des annotations pour décrire les requêtes HTTP, le remplacement des paramètres URL et la prise en charge des paramètres de requête est intégrée par défaut. En outre, il fournit des fonctionnalités pour le téléchargement de requêtes et de fichiers en plusieurs parties.

Nous possédons 3 classes pour récupérer les informations de l’API

Classe *Result*

import com.google.gson.annotations.SerializedName  
  
data class Result(  
 @SerializedName("airing")  
 val airing: Boolean,  
 @SerializedName("end\_date")  
 val endDate: String,  
 @SerializedName("episodes")  
 val episodes: Int,  
 @SerializedName("image\_url")  
 val imageUrl: String,  
 @SerializedName("mal\_id")  
 val malId: Int,  
 @SerializedName("members")  
 val members: Int,  
 @SerializedName("rated")  
 val rated: String,  
 @SerializedName("score")  
 val score: Double,  
 @SerializedName("start\_date")  
 val startDate: String,  
 @SerializedName("synopsis")  
 val synopsis: String,  
 @SerializedName("title")  
 val title: String,  
 @SerializedName("type")  
 val type: String,  
 @SerializedName("url")  
 val url: String  
)

Classe pour la recherche d’une série

import com.google.gson.annotations.SerializedName  
  
data class SearchedAnime(  
 @SerializedName("API\_DEPRECATION")  
 val aPIDEPRECATION: Boolean,  
 @SerializedName("API\_DEPRECATION\_DATE")  
 val aPIDEPRECATIONDATE: String,  
 @SerializedName("API\_DEPRECATION\_INFO")  
 val aPIDEPRECATIONINFO: String,  
 @SerializedName("last\_page")  
 val lastPage: Int,  
 @SerializedName("request\_cache\_expiry")  
 val requestCacheExpiry: Int,  
 @SerializedName("request\_cached")  
 val requestCached: Boolean,  
 @SerializedName("request\_hash")  
 val requestHash: String,  
 @SerializedName("results")  
 val results: List<Result>  
)

Classe pour les Top animes

import com.google.gson.annotations.SerializedName  
  
data class TopAnime(  
 @SerializedName("API\_DEPRECATION")  
 val aPIDEPRECATION: Boolean,  
 @SerializedName("API\_DEPRECATION\_DATE")  
 val aPIDEPRECATIONDATE: String,  
 @SerializedName("API\_DEPRECATION\_INFO")  
 val aPIDEPRECATIONINFO: String,  
 @SerializedName("request\_cache\_expiry")  
 val requestCacheExpiry: Int,  
 @SerializedName("request\_cached")  
 val requestCached: Boolean,  
 @SerializedName("request\_hash")  
 val requestHash: String,  
 @SerializedName("top")  
 val top: List<Result>  
)

**Visuel de l’application**

