

Projet : application musicale

Introduction	2
Les notions	3
L'application	3
L'API	3
L'application	4
Notions	4
Fonctionnement	4
Le design de l'application	5
Page d'accueil	6
Onglet Classements	6
Onglet Recherche	6
Onglet Favoris	6
Détails d'un artiste	7
Détails d'un album	7
Rendu	8
Grille de notation	8

Introduction

Présentation générale :

Vous devrez développer une application mobile utilisant le framework Flutter. Elle se basera sur les données de l'API [TheAudioDB](#), qui fournit des informations autour de la musique, des artistes...

Ce projet sera à réaliser par groupe de 3 personnes.

Les notions

L'application

L'application devra fonctionner sur les smartphones Android et iPhone en mode portrait.
Par simplicité, aucun mode hors ligne ne sera demandé, ni des déclinaisons pour tablettes.

Fonctionnalités minimales

- Page d'accueil avec onglets
- Classements (charts) des singles & albums
- Détails d'un artiste
- Détails d'un album
- Fonction de recherche d'un artiste et par album
- Persistance des favoris

L'API

Vous utiliserez l'API [TheAudioDB](https://theaudiodb.com/) qui permet d'accéder aux informations sur les artistes, albums, classements... Cette API étant "semi-payante", vous pourrez utiliser le token **523532** qui donne accès aux APIs de type "Patreon".

Ex : `https://theaudiodb.com/api/v1/json/523532/album.php?i=112024`

L'énoncé ne vous guidera pas dans le choix [des APIs](#) : ce sera à vous de trouver le ou les plus pertinentes pour chaque écran.

L'application

Notions

Le développement de cette application nécessitera d'utiliser les notions suivantes :

- Conception des interfaces
- BLoC pour la gestion de l'état
- Navigator 2 avec GoRouter pour la gestion des écrans
- Requêtes à l'API avec Retrofit
- Stockage des données avec Hive / Isar

Fonctionnement

L'application sera amenée à faire de nombreuses requêtes vers l'API.

Il est donc important de :

- Ne pas bloquer l'interface utilisateur
- Faire patienter l'utilisateur (ex : afficher un loader)
- En cas d'absence de résultat, afficher un message d'erreur
- En cas d'erreur, afficher un message et permettre de relancer la requête

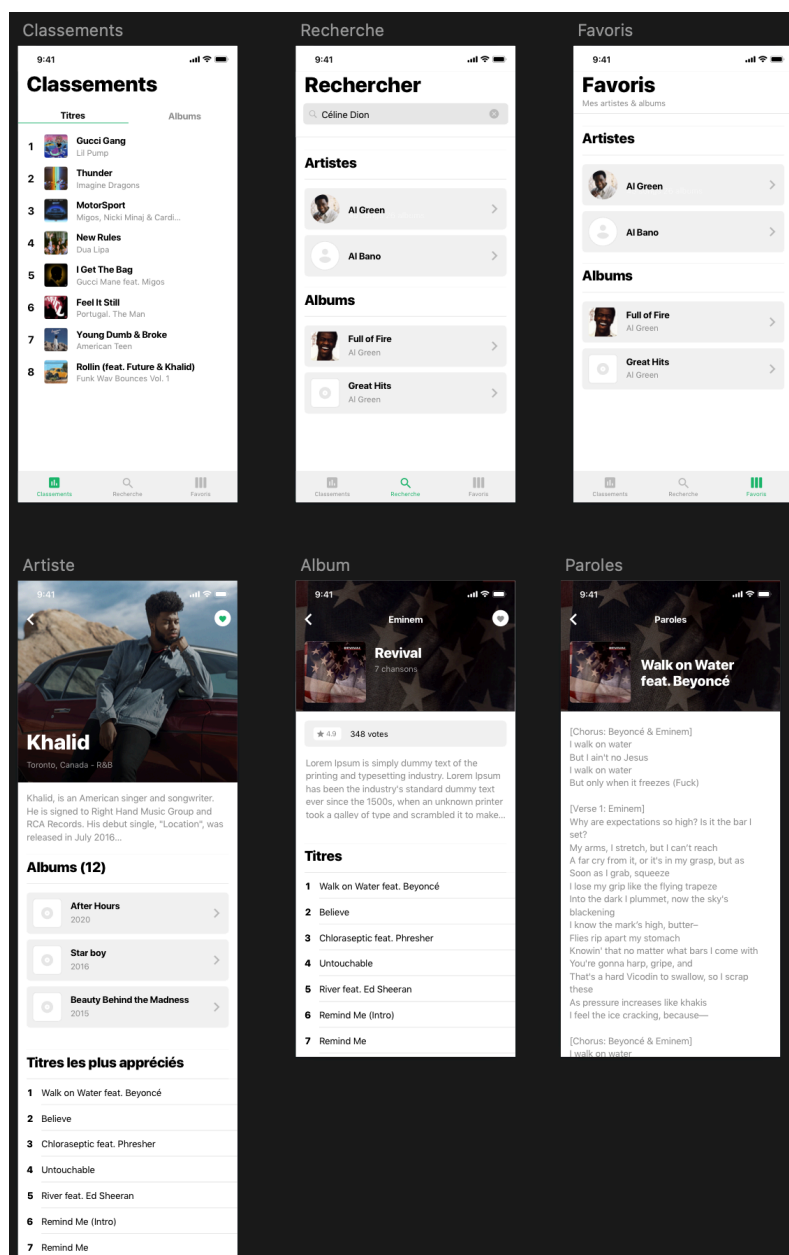
Le design de l'application

Les [différents écrans de l'application](#) vous sont fournis [sur Sketch](#).

- Vous pouvez utiliser la fonction `Inspect` (touche I) pour obtenir plus d'informations.
- [Les icônes](#) sont disponibles au format SVG

La police s'appelle SF Pro et peut-être [téléchargée sur ce lien](#) ou [sur Drive](#).

L'implémentation (et donc le respect) de ce design sera un critère important dans la notation.



Page d'accueil

La page d'accueil affiche trois onglets : les classements, une recherche, les favoris de l'utilisateur. Par défaut, l'utilisateur arrive sur les classements.

Onglet Classements

Les classements permettent de visionner le TOP iTunes pour les États-Unis (les autres données de l'API n'étant plus actualisées) avec les singles, puis les albums.

Le clic sur un artiste ouvrira l'écran de détails.

Onglet Recherche

L'utilisateur pourra saisir une recherche textuelle (ex : Coldplay) et la liste des artistes (ainsi que leurs albums) correspondants s'afficheront alors.

Le clic sur un artiste ouvrira l'écran de détails.

Le clic sur un album ouvrira l'écran de détails.

Deux APIs sont à utiliser : `search` & `searchalbum`.

Onglet Favoris

Sur la fiche d'un artiste, l'utilisateur peut l'ajouter dans ses favoris. Cet écran permet simplement de les lister.

Le clic sur un artiste ouvrira l'écran de détails.

Détails d'un artiste

A partir des informations fournies par l'API, vous devrez fournir le maximum d'informations sur l'artiste. Il faudra par ailleurs afficher la description de l'artiste en fonction de la langue du téléphone, ou en anglais si l'information n'est pas disponible.

Parmi les données affichées, il y aura notamment la liste des albums. Le clic sur un album ouvrira son écran de détails.

Détails d'un album

À partir des informations fournies par l'API, vous devrez fournir le maximum d'informations sur l'album. Il faudra par ailleurs afficher la description de l'album en fonction de la langue du téléphone, ou en anglais si l'information n'est pas disponible.

Rendu

Le projet sera à rendre [sur Boostcamp](#) pour le vendredi 11 avril à 23:59:59.

Vous devrez livrer :

- Le code source de l'application après avoir lancé la commande `flutter clean` (c'est-à-dire sans le dossier `build`)
- Un document présentant les fonctionnalités développées ou non

Grille de notation

Tout code dupliqué sera sanctionné par un zéro pour l'ensemble des personnes du groupe.

Fonctionnalités/Écran	Sous-fonctionnalité	Points
Respect des bonnes pratiques	Import des ressources	1
	Polices, couleurs...	
Requêtes vers l'API	Utilisation de <code>json_serializable</code> + <code>Retrofit</code> & Organisation du code	2
Stockage des favoris	Mise en œuvre	1,5
Changement d'écran	<code>Navigator 2</code> / <code>GoRouter</code>	1
Écran Classements	Implémentation des fonctionnalités demandées	2
	Utilisation de <code>BLoC</code>	0,5
	Respect du design	1,5

Écran Recherche	Implémentation des fonctionnalités demandées	1
	Utilisation de BLoC	0,5
	Respect du design	0,5
Écran Favoris	Implémentation des fonctionnalités demandées	1
	Utilisation de BLoC	0,5
	Respect du design	0,5
Écran Détails d'un artiste	Implémentation des fonctionnalités demandées	2
	Utilisation de BLoC	0,5
	Respect du design	1
Écran Détails d'un album	Implémentation des fonctionnalités demandées	2
	Utilisation de BLoC	0,5
	Respect du design	0,5