

Projet



Application Catalogue Netflix-Like

- > Swift ou React Native ou Flutter
- > API
- > MVC



Groupe de 4 personnes



1

Howard



2

Rajesh



3

Sheldon



4

Leonard

Groupe Inter-TD
autorisé

Application Netflix-Like

Objectifs

Réaliser une application universelle (iPhone/iPad) en Swift 5.0 contenant :

- affichage une vue "principale" d'une liste de titre de film dans un UITableViewController sur le thème de votre choix ou les derniers films récents
- Lorsqu'on sélectionne un film, on accède avec un segue à une vue "détail" où l'on va retrouver différentes informations du film (synopsis, date de sortie, etc.)

Une fois les données téléchargées, l'application est alors consultable en mode hors connecté / mode avion avec CoreData (bonus)

Ergonomie

API films :

<https://www.themoviedb.org/documentation/api>(REST/JSON)

<https://www.themoviedb.org/documentation/api/discover> (EXAMPLES)

Astuces

Pour la Vue "Principale" : utiliser un UITableViewController

Pour la Vue "Détail" : utiliser un UIViewController

Aussi, n'oubliez pas d'utiliser un UINavigationController afin de pouvoir naviguer entre les Vues (Editor > Embed in > Navigation Controller) 😊

Architecture

MVC

Interface

Au moment de la création du projet sur Xcode

Veillez à bien sélectionner STORYBOARD et non SwiftUI

Utiles

L'utilisation de bibliothèques externes sont autorisées

L'utilisation de cocoapods ou de carthage sont autorisées

L'utilisation de Flutter ou React Native sont autorisés

Pour iOS : L'utilisation de SwiftUI n'est pas autorisé

Sujet alternatif

Objectifs

Réaliser une application universelle (iPhone/iPad) en Swift 5.0

contenant :

- affichage une vue "principale" d'une liste d'informations dans un UITableViewController sur le thème de votre choix ou les derniers films récents
- Lorsqu'on sélectionne une cellule, on accède avec un segue à une vue "détail" où l'on va retrouver différentes informations liés à votre sujet

Une fois les données téléchargées, l'application est alors consultable en mode hors connecté / mode avion avec CoreData (bonus)

Ergonomie

API de votre choix

Astuces

Pour la Vue "Principale" : utiliser un UITableViewController

Pour la Vue "Détail" : utiliser un UIViewController

Aussi, n'oubliez pas d'utiliser un UINavigationController afin de pouvoir naviguer entre les Vues (Editor > Embed in > Navigation Controller) 😊

Architecture

MVC

Interface

Au moment de la création du projet sur Xcode

Veillez à bien sélectionner STORYBOARD et non SwiftUI

Utiles

L'utilisation de librairies externes sont autorisées

L'utilisation de cocoapods ou de carthage sont autorisées

L'utilisation de Flutter ou React Native sont autorisés

Pour iOS : L'utilisation de SwiftUI n'est pas autorisé

Bonus



Mode hors connecté

Mise en place et utilisation de CoreData pour la persistance des données

Tests unitaires

Mise en place de tests unitaires via XCTest pour les appels API

 **Astuces**

N'oubliez pas de cocher "Use Core Data" et "Include Tests" lors de la création de votre projet ! 🤗

Ressources

Parser et récupérer les data d'une API

"[How to Parse JSON in Swift 5](#)" de Mark Price

"[Fetching Weather Data with Swift](#)" de Derek Banas

"[JSON data into UITableView with images](#)" de PlanetVeracity

Passer de la data d'un Controller à un autre

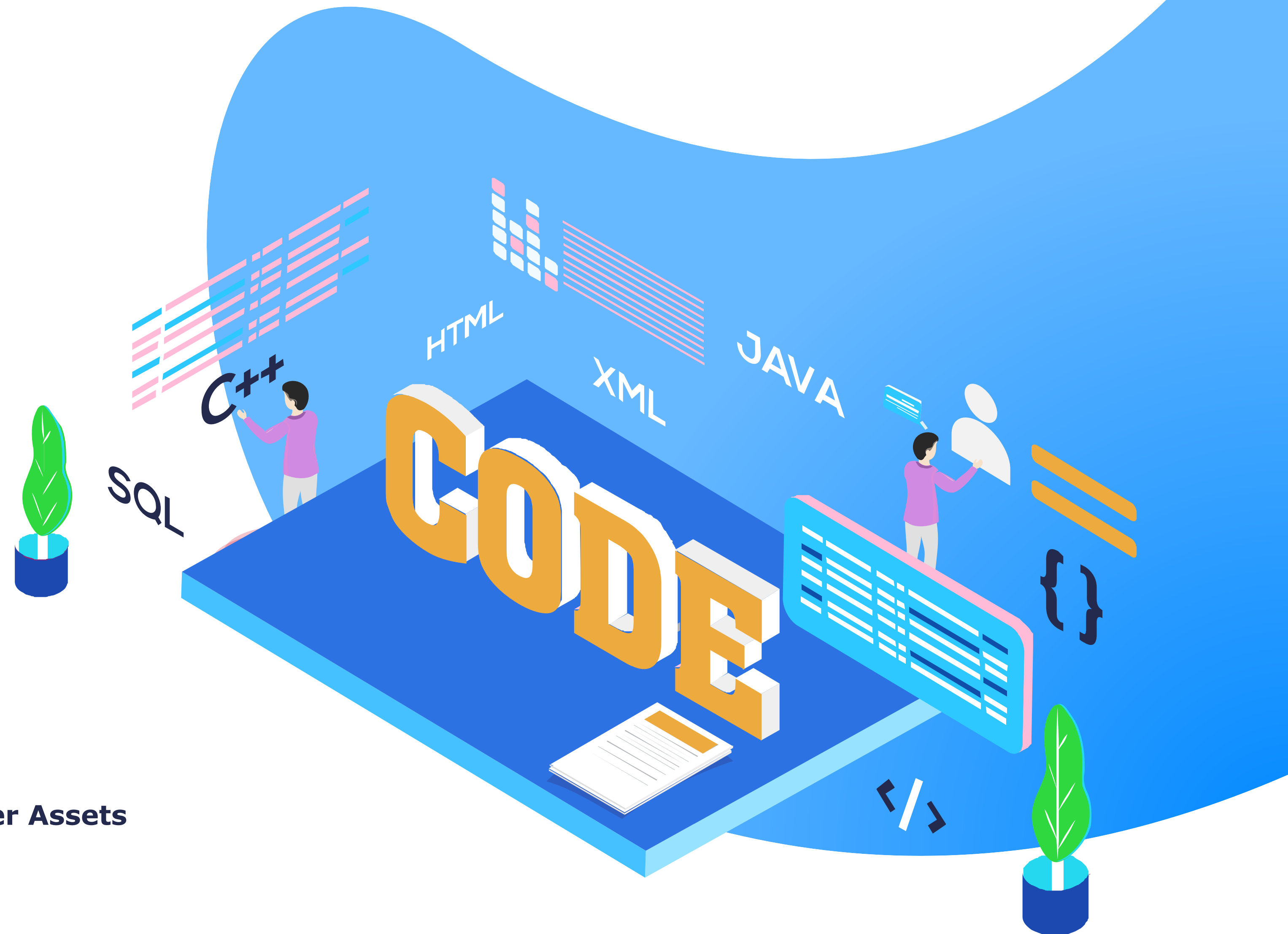
"[Pass Data from TableView and ViewController](#)" de iOS Academy

Être en plein écran lors d'un passage en Segue

"[Full Screen Segues](#)" de Ajay Gandecha

Convertir vos images en x1, x2, x3 pour le dossier Assets

www.resizeappicon.com



⚠ Attention

Attention à bien rendre votre projet sans warning (⚠) et/ou error (❗), sous peine d'avoir des points en moins !

Barème /20 *

Storyboard

Interface

- 1 Utilisation d'un UITableViewController pour la vue "Principale"
- 1 Utilisation d'un UIViewController pour la vue "Détail"
- 1 Utilisation d'un Segue avec Show pour lier les deux vues
- 1 Utilisation de UIImage via Assets.xcassets
- 5 Mise en place de contraintes cohérentes

Controller

- 1 Utilisation d'un UITableViewController
- 1 Utilisation d'un UIViewController
- 1 Les deux controller sont liées à leur vue respective
- 2 Le code est lisible, propre, indenté et commenté

Model

- 2 Appel API
- 2 Traitement des données de l'API (parser)
- 2 Affichage des données de l'API pour chaque vue

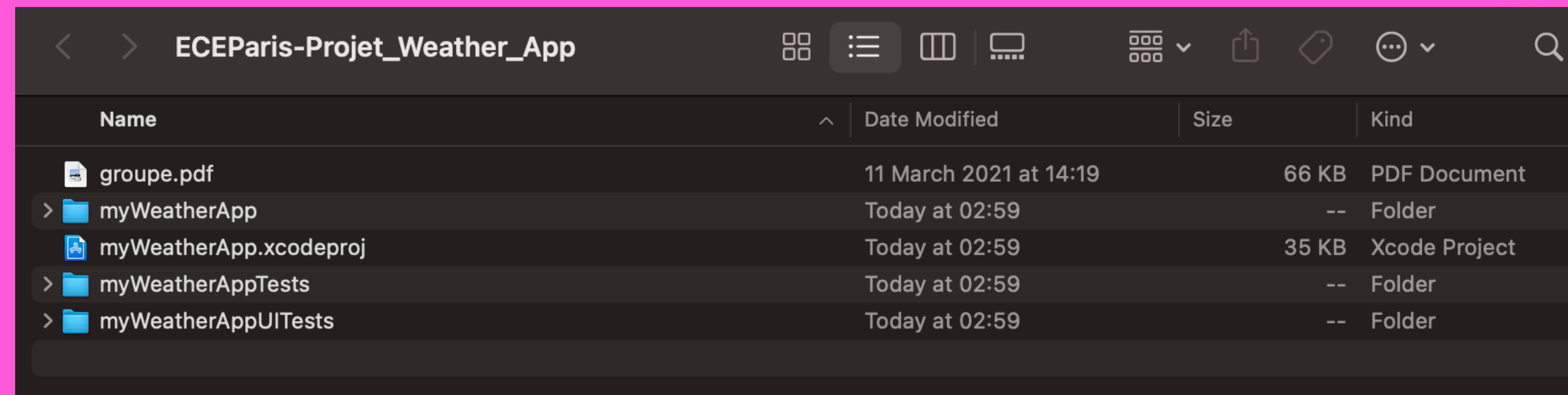
* Sans compter les points bonus 😎

Rendu le 10 avril 2022

⚠ Attention

Pour les chefs de groupe qui effectueront le rendu :
Pensez à bien ajouter un fichier **.PDF** "groupe.pdf" en indiquant les personnes de votre groupe

Merci de me faire parvenir les livrables comme sur l'exemple :



Nom du dossier : "ECEParis-Projet_Netfloux_App" avec dedans :

- L'intégralité du projet Xcode en langage Swift
- Un fichier **.PDF** "groupe.pdf" avec le prénom et nom de chacune des personnes du groupe

Et de compresser le tout dans un fichier **.ZIP**
à l'adresse mail suivante :

danyl.semmache@gmail.com

Avril 2022

Jeu	Ven	Sam	Dim	Lun	Mar	Mer
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7