

<p>Projet Android :</p> <p>Application sur les jeux vidéo</p>

Introduction	2
Les notions	3
L'application	3
L'API	4
Le backend	5
L'application	6
Général	6
Notions	6
Fonctionnement	6
Le design de l'application	7
Page d'accueil	9
Onglet Classements	9
Onglet Recherche	9
Onglet Favoris	10
Détails d'un artiste	10
Détails d'un album	11
Paroles d'une chanson	11
Annexes :	11
Notation	12
Grille de notation	12

Introduction

Présentation générale :

Vous devrez développer une application Android permettant à un utilisateur d'indiquer quels sont ses jeux vidéos favoris à travers l'API de Steam.

L'objectif est de lier un backend de type BaaS (Firebase, Supabase, Appwrite...), permettant de gérer l'authentification, tout comme les choix faits par les utilisateurs.

Ce projet sera à réaliser par groupe de trois personnes.

Les notions

L'application

L'application devra fonctionner sur Android 5.0+ et avoir été développée uniquement en Kotlin. Par simplicité, aucun mode hors ligne ne sera demandé, ni des déclinaisons pour tablettes.

Fonctionnalités minimales

- Écran d'authentification
 - Connexion d'un utilisateur
 - Inscription d'un nouvel utilisateur
 - Récupération du mot de passe
- Accueil avec les jeux du moment
- Liste des jeux de l'utilisateur
- Rechercher un jeu
- Écran de détail d'un jeu
- Liste des avis sur un jeu
- Écran de paramètres pour se déconnecter de son compte
 - > retour à l'écran d'authentification

L'API

Vous utiliserez l'API de Steam.

Pour l'implémenter, vous aurez deux choix :

- Soit l'appeler directement dans l'application
- Soit le faire au niveau backend, offrant ainsi un cache et une expérience commune quelle que soit l'utilisateur connecté

Documentation sur l'API :

<https://steamapi.xpaw.me/#>

ou <https://partner.steamgames.com/doc/store>

Classements des jeux les plus joués :

<https://api.steampowered.com/ISteamChartsService/GetMostPlayedGames/v1/?>

Détails d'un jeu :

<https://store.steampowered.com/api/appdetails?appids=xxx>

Avis sur un jeu :

<https://store.steampowered.com/appreviews/apild?json=1>

Le backend

Vous devrez utiliser un backend de type Backend-As-A-Service comme Firebase, Appwrite... la solution ne vous est pas imposée.

En revanche, le code développé, ainsi que des identifiants pour accéder à la console d'administration devront être fournis.

Sur ce backend, vous devrez gérer :

- **L'authentification** (création, connexion et perte de mot de passe)
Par simplicité le couple email/mot de passe suffira
- **Stocker les données de l'utilisateur**

L'application

Général

- L'application devra fonctionner sur Android 5.0+ (API 21)
- L'application devra être développée en Kotlin
- L'application devra fonctionner sur les smartphones et en portrait
- L'application devra tenir compte de la langue du téléphone
(ex : afficher le contenu en français si le téléphone est en français)

Notions

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

- Conception de vues en XML (LinearLayout, ConstraintLayout...)
- Les ressources
- Les fragments
- Le graphe de navigation
- Les requêtes
- La connexion à un backend

Fonctionnement

Lorsque l'application fera des appels vers Firebase (et éventuellement l'API Steam), il est 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 compréhensible et en rapport avec l'erreur
- En cas d'erreur, afficher un message et permettre de relancer la requête
(ex : Une erreur est survenue... n'est pas un message d'erreur compréhensible)

Le design de l'application

La police est Proxima Nova et [peut-être téléchargée ici](#). L'intégration dans l'application [se fait de cette manière](#).

Les différents écrans de l'application vous seront fournis sous la forme d'images et d'un [fichier Sketch/Lunacy](#). L'implémentation (et donc le respect) du design sera un critère particulièrement important dans la notation.

Authentification

Si un utilisateur lance l'application pour la première fois ou s'il s'est déconnecté, il doit arriver sur l'écran permettant de se connecter.

Écran connexion

L'utilisateur peut saisir son email / mot de passe.

S'il n'a pas de compte, il peut s'en créer un.

S'il a perdu ses identifiants, il peut les redemander.

En cas de succès, ses informations sont récupérées et il est alors redirigé vers la page d'accueil.

Bonus : transférer l'email / mot de passe saisi vers l'écran de création de compte ou l'email uniquement pour la perte de mot de passe

Écran création de compte

L'utilisateur peut saisir son email / mot de passe et le compte est créé aussitôt (pas besoin de validation par email)

En cas de succès, son compte est créé et il est alors redirigé vers la page d'accueil.

Depuis cet écran, le bouton retour le fait revenir vers la connexion.

Écran mot de passe perdu

L'utilisateur saisit son email et il peut recréer un mot de passe.

Peu importe le process ensuite (email, in-app...)

Page d'accueil

Sur la page d'accueil, on peut voir les jeux les plus joués sur Steam. Un clic sur un jeu ouvre son écran de détail.

Écran de détail d'un jeu

À partir des informations récoltées sur Steam, vous affichez les différentes informations et gérez les différentes langues si possible (sinon en anglais).

Le clic sur les captures d'écran doit ouvrir un plein écran avec défilement horizontal.

L'utilisateur peut également voir les avis laissés sur Steam.

L'utilisateur peut liker un jeu et/ou l'ajouter à sa *whishlist*.

Écran du profil du joueur

L'utilisateur peut voir l'ensemble de jeux qu'il a liké, ainsi que ceux de sa wishlist.

Un raccourci pour se déconnecter doit également être proposé.

Notation

Ce projet fera l'objet d'une soutenance (en distanciel ?) le 03 février :

- AL1 : à partir de 11 h 30
- AL2 : à partir de 15 h 45

D'une durée de 12 minutes, vous devrez :

- 1 min : présenter les fonctionnalités développées et celles manquantes
- 5 min : faire une présentation de toutes les fonctionnalités développées
- 6 min : questions / réponses

Ordre de passe : à définir

Le code devra être déposé sur MyGES AVANT le 03 février 11h avec :

- Un lien Git public avec le code Android
- Un lien Git public vers le code du backend
- Des identifiants de connexion à l'administration du backend

Barème de notation

Soutenance : 12/20

Code : 08/20