
Sujet de TP N°2 - Application de Quiz

Date de rendu : 18/06/2024

Modalité de rendu : rendre accessible le code (préférence gestion de sources type Github) ou sinon ZIP par email, drive etc.

Notation:

- 12 pts sur la partie fonctionnelle (l'application répond au besoin)
- 8 pts sur la partie technique (architecture, qualité du code, crash)

VUE D'ENSEMBLE

Conception et développement d'une application **Android en Kotlin** proposant à un utilisateur de répondre à des Quizzes. L'application devra implémenter les fonctionnalités suivantes:

- Écran de connexion où l'utilisateur choisit son pseudo, une fois renseigné, il est redirigé vers l'écran principal de l'application.
- Sur l'écran principal, l'utilisateur a le choix entre plusieurs catégories de quizzes. (jeux-vidéos, cinéma, art et littérature, musique, culture générale, sport etc.)
- Après avoir sélectionné une catégorie, l'utilisateur répond aux questions les unes après les autres.
- A tout moment, l'utilisateur connaît son nombre de réponses correctes et fausses.
- L'utilisateur possède un temps limité sur chaque question, à la fin du temps imparti, la question compte comme une réponse fausse.
- A la fin d'un Quizz, une page des scores est présentée avec les pseudos et le nombre de réponses correctes.
- Une fois l'écran des scores acquitté, l'utilisateur est redirigé vers le premier écran de connexion.

OBJECTIFS

1. Le premier objectif est de restituer les différentes notions de la conception et du développement d'une application Android.
2. Le second objectif est de sensibiliser à la production d'une application "professionnelle" que ça soit : par l'architecture logiciel établie ; la qualité du code ; la gestion du code source ; la sensibilité à l'expérience utilisateur.

CARACTÉRISTIQUES

L'application doit disposer des caractéristiques suivantes :

- **Inscription / Connexion** : L'utilisateur doit pouvoir renseigner un pseudo pour accéder aux quizzes. Une fois renseigné, l'utilisateur est redirigé vers la page de choix de catégorie.
- **Navigation** : Un ou des composants de navigation permettent à l'utilisateur de naviguer dans les différents écrans de l'application. ⚠ Je ne veux pas voir une seule activité ou un seul fragment qui gère toute l'application, elle doit être découpée judicieusement.
- **Catégories des quizzes** : Les catégories sont libres. Pour ce premier TP elles seront en dur dans votre code pour simplifier le développement. Idem pour les questions de chaque catégorie.
- **Déroulement d'un quiz** : Quand l'utilisateur a sélectionné sa catégorie, son quiz est généré aléatoirement en prenant 10 questions parmi les questions enregistrées.
- **Page des scores** : À la fin du Quiz, le score est enregistré dans les *"shared preferences"* pour la catégorie et le pseudo donné. Cette page restitue les meilleurs scores.
- **Architecture** : L'architecture de l'application est libre de choix.
- **Base de données** : L'application doit avoir une base de données créée avec la bibliothèque Room. Cette base de données va contenir les différentes questions et catégories de quizz. Structure de données libre.

[BONUS]

- **Requêtes HTTP** : L'application utilise des requêtes HTTP pour récupérer des quizzes en ligne. Au démarrage ou lors du choix d'une catégorie, les questions sont récupérées du serveur et enregistrées dans la base de données. [Quiz API](#)

Conseils pour les requêtes :

- Regardez du côté des librairies [OkHttp](#) ou [Retrofit](#) pour les appels API.
- Et [Moshi](#) ou [Kotlinx Serialization](#) pour désérialiser du Json en objet.

Vous pouvez bien sûr le faire avec d'autres librairies ou même à la main si ça peut vous rendre heureux.

Ressources

-
- [Kotlin cheatsheet](#)
 - [SharedPreferences](#)
 - [Room](#)

Pensez à vérifier la robustesse de votre application, l'idée étant d'avoir au minimum une application qui ne crash pas. Essayez quelques cas aux limites / ce qu'un utilisateur pourrait faire avec votre application :

- Changement de configuration (rotation...)
- Passe en background / revenir sur l'application
- etc.