
Sujet de TP N°2 - Application de Quiz

Date de rendu :

Modalité :

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 Quiz, 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.

NEW

- **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.

TP

1. **Affichage** : Partir du précédent TP et terminer la mise en place des vues, activités, fragments.
2. **Configuration** : Ajouter les dépendances de Room dans le fichier *build.gradle*
3. **Entity** : Créer les data class kotlin et ajouter les annotations *d'Entity* pour que Room génère la structure de données.
4. **DAO** : Créer l'interface *DAO* avec les différentes méthodes nécessaires au bon déroulement d'une partie de Quiz.
5. **Base de données** : Créer une classe abstraite qui hérite de *RoomDatabase*, y ajouter les méthodes pour avoir un singleton de l'instance de base de données et pouvoir récupérer le ou les *DAO*.

-
6. **Repository** : Créer une classe *QuizRepository*, cette classe servira de seul point d'accès aux données, que ce soit pour la base de données ou pour les futurs appels API.

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.