

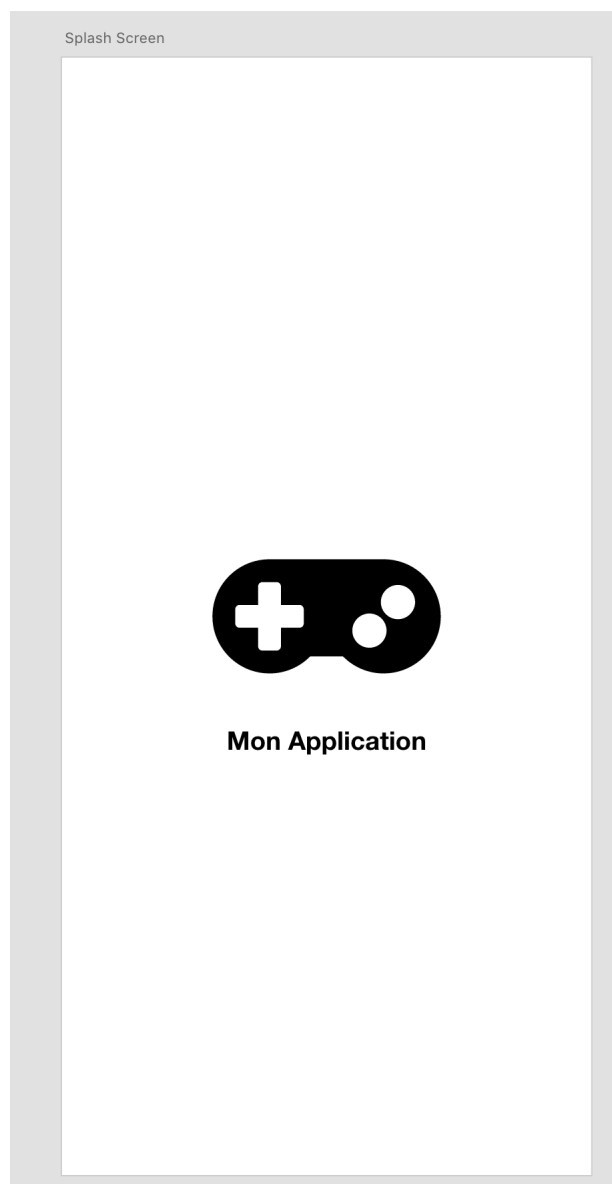
Projet Android - Kotlin, ViewModel LiveData & Room

L'objectif de ce projet est de créer une application mobile offline sur android utilisant Jetpack navigation component, viewModels, LiveData et Room.

L'application est un mini jeu consistant à poser des questions à l'utilisateur et de le laisser deviner la réponse.

L'application se constitue ainsi de 5 écrans.

1. Écran splash :



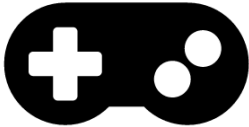
Point d'entrée de l'application, l'écran splash est définie généralement dans sa partie visuelle par le logo et le nom de l'application, un indicateur de chargement peut également être ajouté ([voir le composant](#))

C'est dans cet écran que l'initialisation de l'application s'effectue (chargement des éléments de la base de données locale, vérification de l'authentification, etc)

2. Écran login

Login screen

Mon Application



Identifie toi avec ton pseudo

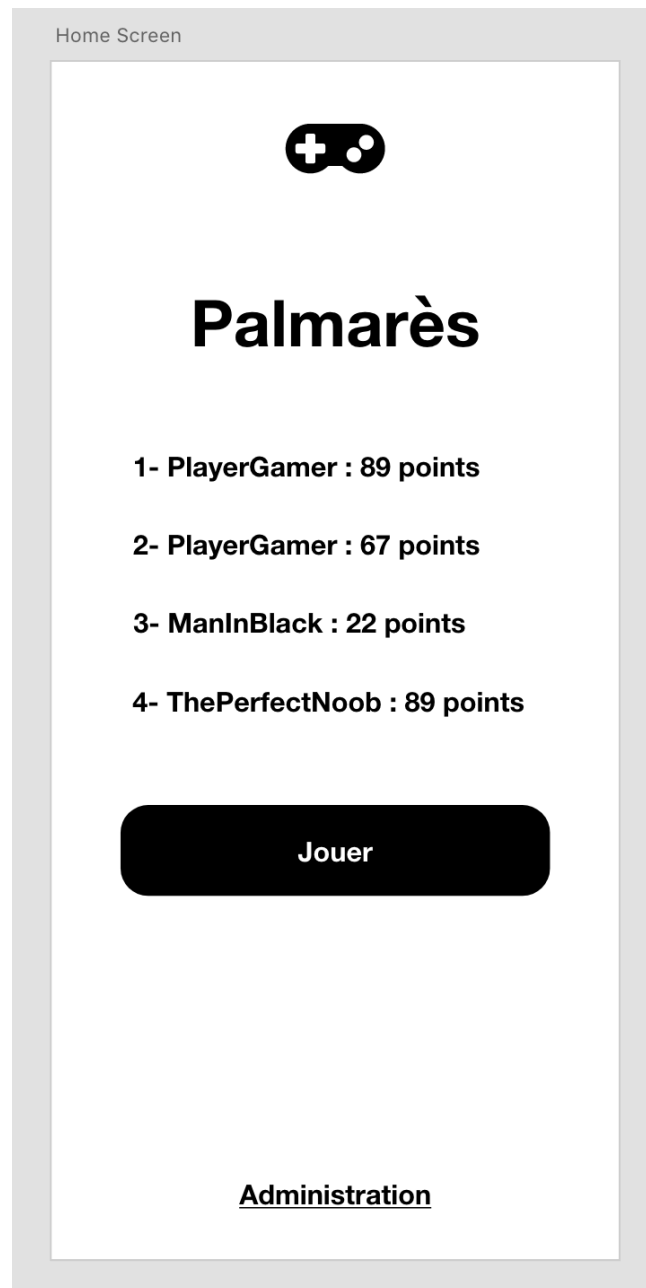
Pseudo

Commencer

Constitué généralement d'un ou plusieurs champs d'édition, l'écran de login permet d'identifier l'utilisateur (joueur)

Le bouton “commencer” permet de sauvegarder le nom de l'utilisateur et de naviguer dans la suite de l'application

3- écran Home



L'écran home est composé d'une liste représentant le palmarès des joueurs (les 4 meilleurs) et de deux boutons

Le premier bouton navigue vers l'écran du jeu. et le deuxième navigue vers l'écran d'administration

4- écran jeu

Home Screen – 1

Score **9** Pt

Devine le nom du chat
et gagne des points

10 chat sur 12

mossiai

Réponse

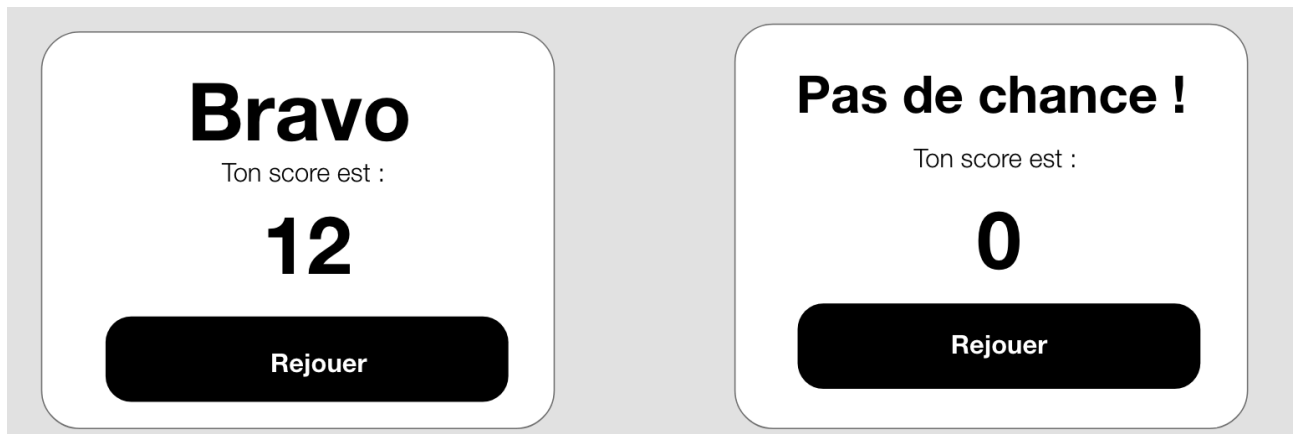
Deviner

Passer

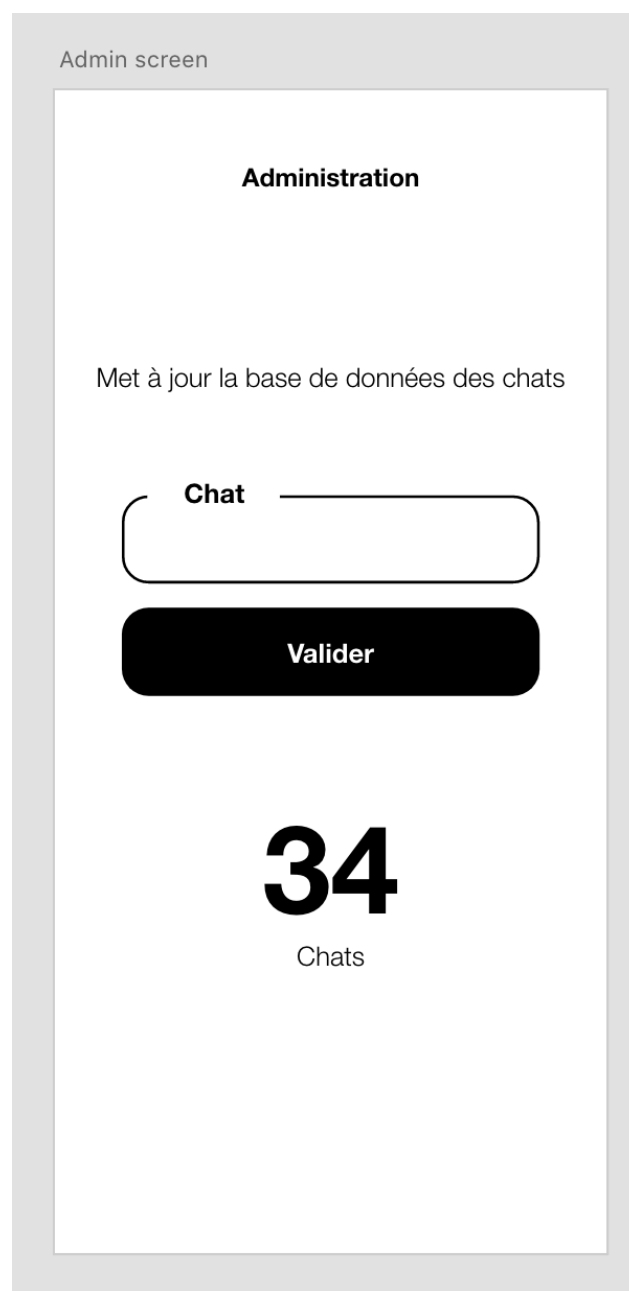
L'écran jeu est composé d'un texte désordonné représentant la race du chat, d'un champ de texte pour recueillir la réponse du joueur. D'un premier texte contenant le score et d'un deuxième texte contenant le nombre de réponses à trouver.

Deux bouton sont placé sous le champs texte pour valider ou ignorer la race du chat en cours.

Une boite de dialogue est affichée à la fin de l'épreuve affichant le score



5 - écran administration



L'écran d'administration contient un champ texte. Chaque chat ajouté est sauvegardé dans la base de données et ajouté dans la liste des chats qui apparaîtra dans l'écran du jeu

Consignes

1. S'inspirer des écrans précédents pour créer sa propre application jeu sur android
2. Créer un Nouveau projet android (ou mettre à jour un projet présent)
3. Créer l'activité principale de l'application et les fragments nécessaires
4. Développer l'interface graphique des fragments
5. Mettre en place une navigation avec jetpack Navigation component en créant et développant le fichier nav_graph
6. Mettre en place les viewModels adéquats : authViewModel (pour gérer le username) et gameViewModel pour gérer toute la partie logique du jeu
 1. Voir la documentation pour partager les données d'un viewModel entre plusieurs fragments ([documentation](#))
7. Mettre en place la LiveData et les observateurs nécessaires dans les écrans splash, jeu, home et admin
8. Sauvegarder et persister les données en utilisant [Room](#)

Recommandations :

- Utiliser Git pour le contrôler de la version en ajoutant une nouvelle branche pour chaque étape.
- Pousser le code vers un repertoire gitlab ou github
- M'ajouter dans le repertoire (gitlab : yelbehi@bewizyu.com, GitHub : youssef.elbehi@gmail.com)
- Pour définir une liste dynamique, [RecyclerView](#) peut être utilisé: voir la documentation et le [tutoriel](#)
- [Lien du cours](#)
- [Documentation room](#)
- [Exemple Room + Coroutine + viewModel + LiveData](#)