



Compte Rendu TEA 2 PMR

Réalisé par :

Abdellah CHIOUA

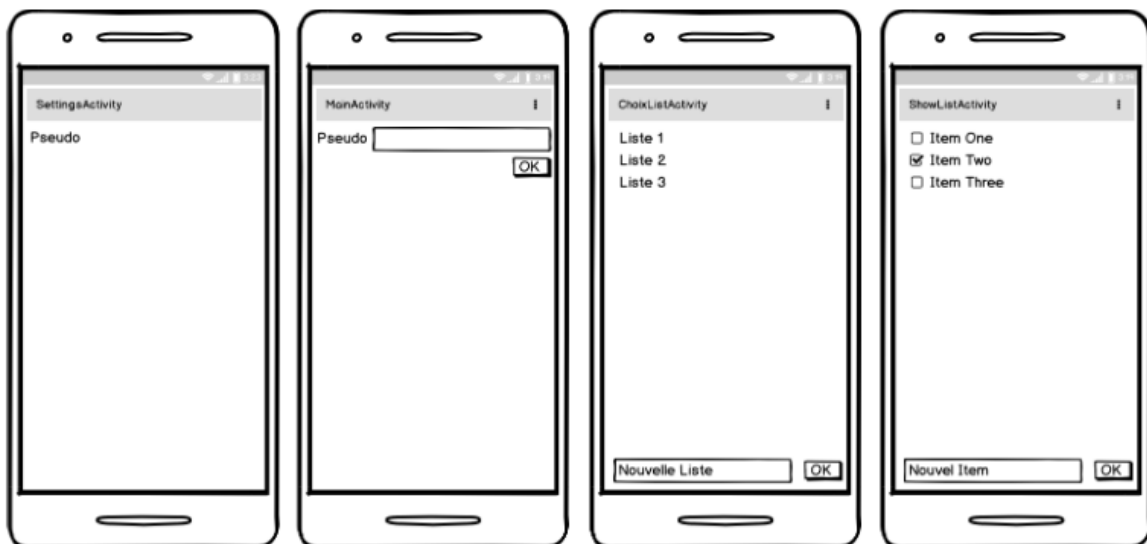
Achraf QIYAOU

Introduction :

L'objectif de ce TEA2 consiste à connecter notre application Storyboard développée lors du premier TEA, à une API en ligne. Nous cherchons à établir cette liaison en utilisant des détails supplémentaires et une analyse plus approfondie.

Pour simplifier notre code et bénéficier de fonctions déjà existantes pour consommer une API et aussi gérer les requêtes de manière appropriée et gérer le multithreading, nous avons décidé d'utiliser la bibliothèque Volley. Elle offre des fonctionnalités qui faciliteront notre travail.

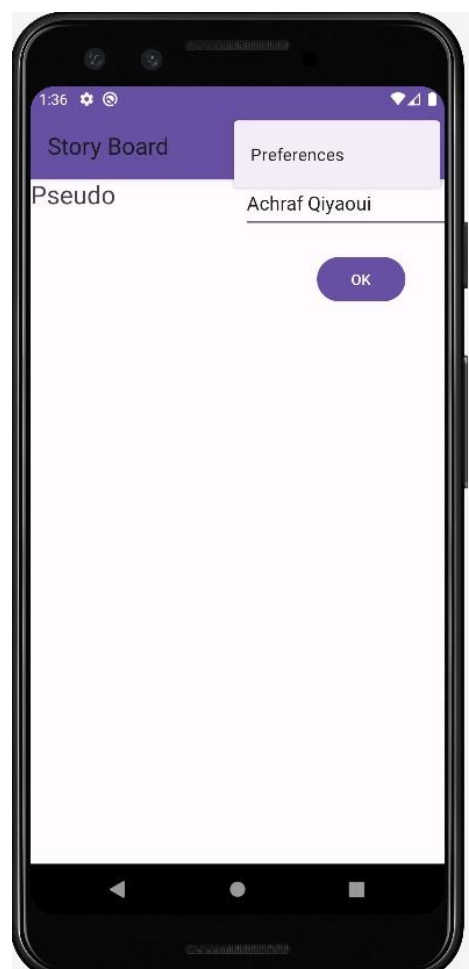
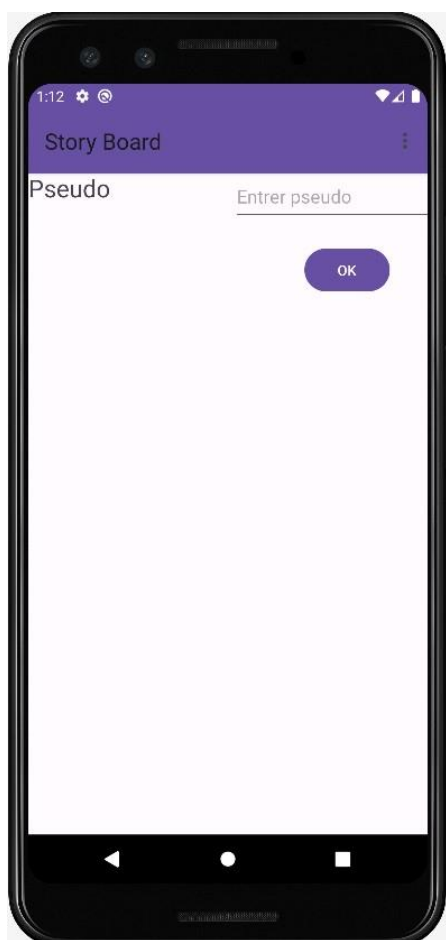
Storyboard



Implémentation :

MainActivity :

Nous avons pu satisfaire toutes les exigences ce TEA, à l'exception de l'implémentation de l'en-tête (header), nous avons choisi de stocker le hachage (hash) en tant que chaîne de caractères dans les préférences au début du développement, car nous n'étions pas familiarisés avec le concept d'en-tête à ce moment-là.



:

SettingsActivity :

Nous avons programmé le layout de cette activité à partir du fichier XML correspondant (activity_settings.xml) de manière à respecter les contraintes mentionnées dans l'énoncé:

- Héritage de PreferenceActivity.
- Le champ "pseudo" est affiché par défaut.

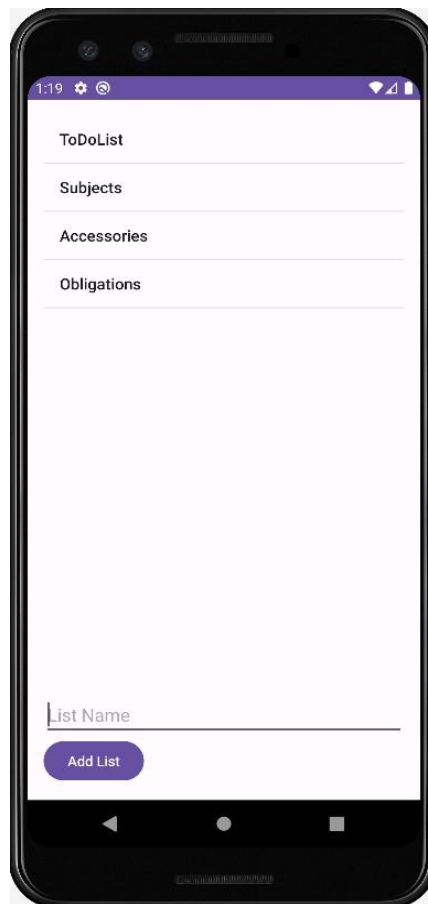
Nous avons également ajouté quelques fonctionnalités qui sont généralement présentes dans l'activité Settings de chaque application:

- Lock Screen.
- Fréquence des rappels des mises à jour.



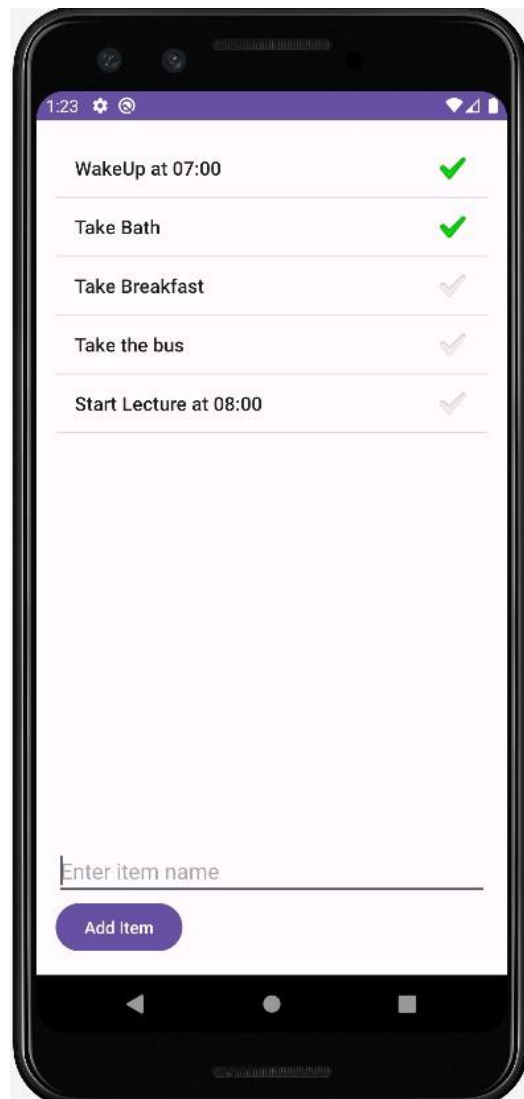
ChoixListActivity :

Nous avons implémenté l'activité ChoixListActivity en utilisant des ListView/RecyclerView et des Adapters pour afficher la liste des listes de l'utilisateur dont le pseudo a été saisi. Cette activité nous permet de nous connecter à l'API afin de récupérer les listes de l'utilisateur et de les afficher de manière appropriée. Lorsque l'utilisateur clique sur une liste, l'activité ShowListActivity est affichée, en lui transmettant l'indice de la liste sélectionnée. De plus, nous avons ajouté la fonctionnalité permettant à l'utilisateur d'ajouter une nouvelle liste à son compte. Ainsi, nous avons étendu les fonctionnalités de notre application pour permettre la gestion complète des listes d'utilisateur.



ShowListActivity:

Nous avons développé l'activité ShowListActivity pour afficher les éléments des listes de l'utilisateur dont le pseudo a été saisi. Cette activité permet à l'utilisateur de visualiser les éléments de la liste, de les cocher ou décocher selon leur état et d'ajouter de nouveaux éléments à la liste. Cependant, malheureusement, nous n'avons pas pu établir la connexion avec l'API dans cette partie de l'application. Par conséquent, les modifications apportées à la liste ne seront pas synchronisées avec l'API en temps réel. Cela reste une fonctionnalité à implémenter dans une future version de l'application.



Conclusion:

Nous avons réussi à mettre en place les différentes activités de l'application, bien que certaines ne correspondent pas exactement aux spécifications de l'énoncé initial. Durant le développement de l'application, nous avons utilisé le langage Kotlin et nous avons exploité les concepts appris en cours, ainsi que des ressources en ligne, pour surmonter les défis rencontrés. Un des apprentissages clés de ce projet a été la mise en place de la connexion à une API en ligne, ce qui nous a permis de développer une meilleure compréhension de ce processus et de renforcer nos compétences dans ce domaine.