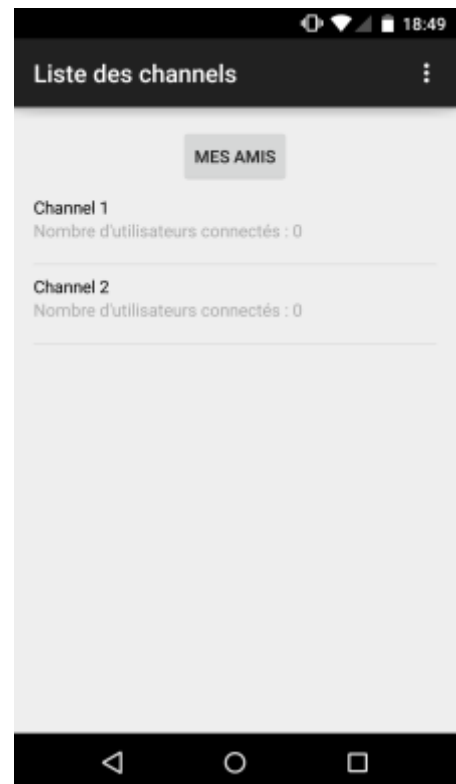


TP Channel messaging (suite)

But du TP : Gérer les bases de données en permettant à un utilisateur d'envoyer des messages privés

I. Ajout de l'activité de la liste d'amis

Dans la liste des channels, ajouter un bouton permettant d'afficher une nouvelle activité qui contiendra la liste des amis.
(cf image à droite)



II. Création de la base de données

En s'aidant du cours, créer les 2 classes (une première nommée FriendsDB héritant de SQLiteOpenHelper et une seconde nommée UserDatasource) qui permettront de créer et d'utiliser la base de données. Cette base de données contiendra une table unique. Elle est décrite ci-dessous :

User

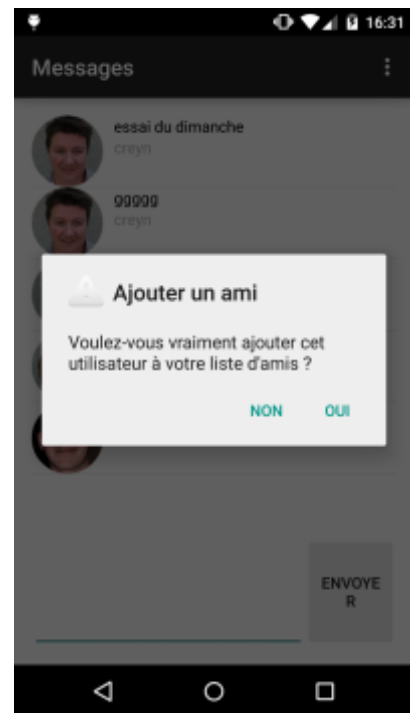
Propriétés :

- userID INTEGER
- username TEXT
- imageUrl TEXT

III. Ajout d'un ami à la base de données

Faire que lorsque l'on appuie sur un message d'une personne autre que soi dans la liste des messages d'un channel provoque le dialogue suivant :

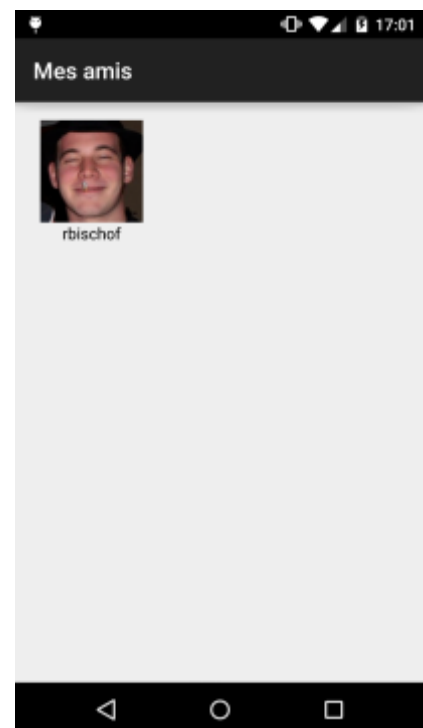
Pour créer un dialogue, il faudra utiliser un AlertDialog (cf [ici](#))



IV. Affichage des amis

Afficher dans l'activité de la liste d'amis une gridview qui contiendra les différents amis de la base de données. Pour faire cela, il vous faudra créer un adapter personnalisé.

Faire que lors du click sur un élément, une nouvelle activité qui affichera les messages s'affiche.



V. Activité des messages privés

Afficher dans une listview les différents messages récupérés par la méthode du webservice « getmessages » qui prend pour paramètre un userid. Une ligne aura à gauche l'image de l'utilisateur distant et à droite l'image de l'utilisateur de l'appareil. Les nouveaux messages seront marqués en gras.

Ajouter un bouton et un edittext pour pouvoir envoyer un message à un destinataire.

