

# TP SQL

## MY Business Directory

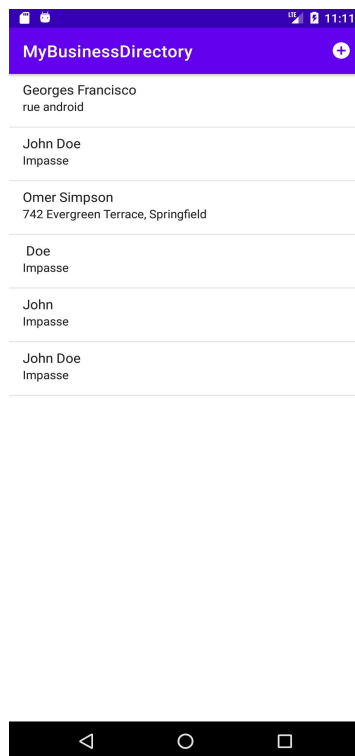
### I. Présentation générale

Ce TP va permettre de voir la gestion d'une base SQL sur la plateforme android afin de sauvegarder une liste de contacts, ainsi que la découverte de nouveaux concepts généraux android, tels que :

- gestion d'une liste
- gestion d'une toolBar et de son menu associé

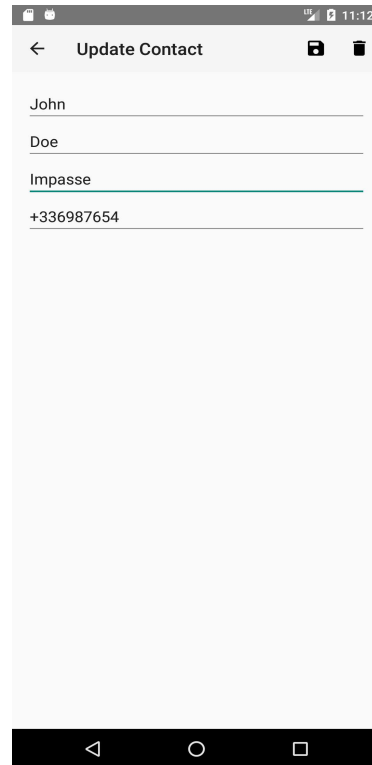
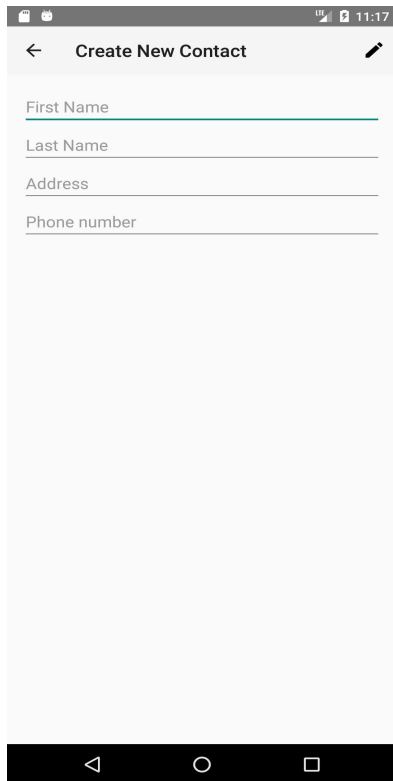
L'application s'articulera sur 2 activités :

La premiere activité nous permet de visualiser la liste des contacts présents dans la base SQL



La seconde activité disposera de 2 modes :

- **Mode création de contact** : lorsque l'on clique sur le bouton + (create) depuis la première activité
- **Mode modification de contact**: lorsque l'on clique sur un contact existant depuis la liste de contacts présentée dans la première activité



## II. Etapes de réalisation

- Création d'une nouvelle application avec le wizard d'Android Studio (sélectionnez Empty Activity)
- Ouvrir le layout de votre activité principale et y placer une ListView qui prendra tout l'écran (vous pouvez également remplacer le layout principale par un LinearLayout)
- Création d'une classe BusinessContact qui contiendra les informations d'un contact à savoir :  
contactId (utile pour plus tard avec la base SQL)  
firstName  
lastName  
address  
phoneNumber  
Vous pouvez créer tous les getter/setters en utilisant Android Studio : click droit → generate → Getter and Setter → sélectionnez toutes les données membres souhaitées
- Dans l'activité vous récupérez le contrôleur de la ListView auquel vous allez lui donner un adaptateur pour gérer l'affichage des éléments de liste ( avec la méthode setAdapter)

C'est adaptateur est une classe que vous devrez créer dans un fichier à part, qui hérite de BaseAdapter. Chaque item de la liste affichera 2 textes l'un au dessus de l'autre.

Soit vous créez vous-même le layout relatif soit vous pouvez réutiliser celui d'android ( android.R.layout.simple\_list\_item\_2 )

- Gestion d'un menu dans l'activité principale :  
Surchargez la méthode onCreateOptionsMenu dans laquelle vous ferez un inflate d'un layout de menu que vous aurez créé préalablement  
Souris : New → Android Resource Directory → Resource Type = menu  
Puis vous aurez accès à : New → Menu Resource File  
Ajouter un item dans ce menu qui va nous permettre d'ouvrir une seconde activité en mode ajout, donc y placer une icône +  
Pour se faire, il est possible de générer des svg avec android studio ;  
Souris : New → Vector Asset → clic sur « clip art » et recherchez un icon +  
Vous pouvez l'assigner à l'entrée de menu avec le paramètre : android:icon  
Vérifiez le bon affichage de l'icône  
Une fois que l'affichage effectué, il vous faudra gérer le clic sur le bouton du menu afin d'ouvrir la seconde activité. Pour se faire, surchargez la méthode onOptionsItemSelected et vérifiez que l'item à l'origine du clic est celui du bouton add en testant son id avec la méthode getItemId() (évidemment vous en profiterez pour créer une seconde activité que l'on pourra appeler CreateContactActivity)

- Création d'une classe qui va gérer la base SQL (MyBusContactDbHelper )

Cette classe hérite de SQLiteOpenHelper

Il est conseillé de créer les constantes de classe suivante pour gérer toutes les requêtes :

DATABASE\_NAME

TABLE\_CONTACT

COLUMN\_CONTACT\_ID

COLUMN\_CONTACT\_FIRSTNAME

COLUMN\_CONTACT\_LASTNAME

COLUMN\_CONTACT\_ADDRESS

COLUMN\_CONTACT\_PHONE

COLUMN\_CONTACT\_ID\_POSITION

COLUMN\_CONTACT\_FIRSTNAME\_POSITION

COLUMN\_CONTACT\_LASTNAME\_POSITION

COLUMN\_CONTACT\_ADDRESS\_POSITION

COLUMN\_CONTACT\_PHONE\_POSITION

Dans cette classe nous aurons :

- le constructeur aura le rôle de créer la base SQL
- la méthode onCreate devra créer la table CONTACT dans la base SQL
- la méthode onUpgrade devra gérer les upgrade de la table (changements de version)  
→ nous ferons un simple DROP TABLE dans ce cas
- la méthode addContact(BusinessContact contact) permet de créer un nouveau contact dans la base SQL
- la méthode getContact(int id) retourne le contact de la base SQL correspondant à l'identifiant donné
- la méthode getAllContacts() retourne tous les contacts de la base SQL
- la méthode getContactsCount() retourne le nombre de contacts disponibles dans la base SQL
- la méthode updateContacts(BusinessContact contact) permet de modifier un contact existant de la base SQL
- la méthode deleteContact(BusinessContact contact) permet de supprimer un contact existant de la base SQL
- la méthode createDefaultContactsIfNeed() permet de peupler la base SQL en utilisant addContact afin d'avoir rapidement des contacts à afficher dans l'activité principale

Remarque : vous pouvez choisir d'implémenter toutes les méthodes maintenant ou de faire celle qui sont nécessaire au fur et à mesure du projet

- Dans la méthode onCreate de l'activité principale instancier MyBusContactDbHelper afin de créer la base SQL et d'avoir la possibilité par la suite de gérer la base et appeler la méthode createDefaultContactsIfNeed pour avoir des contacts à afficher
- Récupérez tous les contacts de la base et vérifiez le bon affichage de ceux-ci

## Gestion de l'ajout d'un contact : → activite CreateContactActivity

- Préparer le layout de cette activité afin d'afficher les 4 informations d'un contact
- Ajouter un Toolbar dans le layout et changer le titre de l'activité dans la methode onCreate en appelant setTitle de la Toolbar ( pour gérer sa propre toolbar il faut désactiver celle par défaut dans la declaration dans le manifest  
`android:theme="@style/Theme.AppCompat.Light.NoActionBar" )`
- Afficher la fleche de retour dans la toolbar avec  
`getSupportActionBar().setDisplayHomeAsUpEnabled()` et  
`getSupportActionBar().setDisplayHomeAsUpEnabled()`
- De la même manière que précédemment gérer le menu de l'activité en y ajoutant une icône de création
- Sur clic du bouton create, vous ajoutez un nouveau contact dans la base SQL en utilisant MyBusContactDbHelper et fermer l'activité courante
- Vérifiez que le contact apparaît bien dans la liste de l'écran principale

## Gestion de l'update d'un contact :

- Gérer le clic sur une entrée de la liste des contacts dans l'écran principale, le clic ouvrira l'activité CreateContactActivity en lui donnant en paramètre le contact sélectionné.
- Dans l'activité CreateContactActivity récupérer le contact en question et modifier le titre en « Update contact » et afficher toutes les données du contact dans les champs adéquats
- Modifier le menu afin d'avoir non plus l'icon create mais 2 icones update et delete
- Gérer les nouveaux boutons du menu et verifiez le comportement global

*Félicitations si vous êtes parvenu à gérer  
tous les cas de cette application*