

Labo3 : Créer un Système de Gestion des Étudiants avec Images

Objectif : Créez une application Android permettant de gérer une liste d'étudiants avec des images en utilisant une base de données SQLite. L'application doit afficher les étudiants dans une ListView via un ArrayAdapter et permettre les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer), y compris la gestion des images.

Exigences :

1. Modèle Étudiant :

- Chaque étudiant doit posséder les propriétés suivantes :
 - id : un entier auto-incrémenté (clé primaire).
 - nom : le nom de l'étudiant (String).
 - âge : l'âge de l'étudiant (Integer).
 - département : le département de l'étudiant (String).
 - photo : une référence vers la photo de l'étudiant (par exemple, le chemin URI d'une image).

2. Configuration de la Base de Données :

- Utilisez une base de données SQLite pour stocker les informations des étudiants.
- Créez une classe StudentDatabaseHelper qui étend SQLiteOpenHelper.
- Implémentez les méthodes pour chaque opération CRUD :
 - **Créer** : ajouter un nouvel étudiant avec sa photo dans la base de données.
 - **Lire** : récupérer tous les étudiants et leurs images depuis la base de données.
 - **Mettre à jour** : modifier les informations et la photo d'un étudiant.
 - **Supprimer** : supprimer un étudiant via son id.

3. Interface Utilisateur :

- Utilisez une ListView pour afficher la liste des étudiants.
- Chaque élément de la liste doit inclure :
 - Le nom de l'étudiant.
 - Son âge et département.
 - Sa photo (utilisez un ImageView).
- Utilisez un ArrayAdapter pour lier les données des étudiants à la ListView.
- Fournissez un formulaire pour ajouter de nouveaux étudiants et pour modifier les étudiants existants, avec une option pour sélectionner ou capturer une image.

4. Gestion des Images :

- Permettez à l'utilisateur de sélectionner une image depuis la galerie ou de prendre une nouvelle photo lorsqu'il ajoute ou modifie un étudiant.
- Enregistrez l'URI de l'image dans la base de données SQLite (vous pouvez utiliser le type String pour stocker l'URI).
- Affichez l'image dans l'ImageView de chaque élément de la ListView.

5. Opérations CRUD :

- **Ajouter** : Lorsque l'utilisateur entre les informations de l'étudiant et sélectionne une photo, ajoutez l'étudiant à la base de données et actualisez la ListView.
- **Modifier** : Lorsqu'un élément de la ListView est longuement pressé, permettez à l'utilisateur de modifier les informations et la photo de l'étudiant. Après la mise à jour, sauvegardez les changements dans la base de données et actualisez la liste.
- **Supprimer** : Lorsqu'un élément est pressé, affichez une boîte de dialogue de confirmation pour supprimer l'étudiant. Si l'utilisateur confirme, retirez l'étudiant de la base de données et mettez à jour la ListView.

Conseils :

- Pour gérer les images, utilisez un URI pour stocker le chemin de l'image dans la base de données SQLite. Pour l'affichage, chargez l'image en utilisant cet URI.
 - Utilisez Intent.ACTION_PICK pour permettre aux utilisateurs de choisir une image dans la galerie.
 - Utilisez des objets Cursor pour récupérer les données de SQLite, y compris l'URI de l'image.
 - Après chaque opération CRUD, mettez à jour l'ArrayAdapter pour actualiser la ListView.
-

Fonctionnalités Exemple :

- **Ajouter un Étudiant** : Saisissez le nom, l'âge, le département, et sélectionnez une photo. Cliquez sur "Ajouter" pour stocker les données de l'étudiant dans la base de données SQLite.
- **Modifier un Étudiant** : Appuyez longuement sur l'entrée d'un étudiant pour ouvrir un formulaire de modification, changez les champs et sélectionnez une nouvelle photo si nécessaire, puis enregistrez les changements.
- **Supprimer un Étudiant** : Appuyez sur le nom d'un étudiant pour confirmer sa suppression, puis retirez-le à la fois de la base de données et de la ListView.

Bonus :

Ajoutez une fonctionnalité **Recherche** pour filtrer les étudiants par nom ou département.

Cette version de l'exercice permettra à vos étudiants de se familiariser avec l'utilisation de SQLite pour gérer des références d'images, de manipuler les données avec ListView et ArrayAdapter, et d'implémenter les opérations CRUD avec des images dans une application Android.