Projet Flutter : Application de Gestion des Utilisateurs (CRUD) -2

Projet Flutter : Application de Gestion des Utilisateurs (CRUD) -2

1. **Description du Projet**
   - Dans ce projet, vous allez continuer à développer votre application de gestion d'utilisateurs en utilisant Flutter et SQLite. Cependant, cette fois-ci, nous allons ajouter une nouvelle fonctionnalité importante : la possibilité d'associer des images aux utilisateurs.
2. **Prérequis**
   - Avant de commencer ce projet, vous devez avoir terminé avec succès le projet précédent, qui consistait à créer une application de gestion d'utilisateurs avec des opérations CRUD de base. Assurez-vous que vous avez mis en place la structure de base de l'application, y compris la base de données SQLite et les opérations CRUD.
3. **Objectifs du Projet**
   - Intégrer la gestion des images : Vous allez ajouter un nouvel attribut image à votre modèle User pour stocker le chemin de l'image associée à chaque utilisateur.
   - Mettre à jour l'interface utilisateur : Vous allez mettre à jour les écrans d'interface utilisateur existants pour permettre aux utilisateurs d'ajouter et de visualiser les images des utilisateurs. Cela peut inclure l'ajout de boutons de sélection d'image, d'espaces pour afficher les images, etc.
   - Mettre à jour les opérations CRUD : Vous allez modifier les opérations CRUD dans le service utilisateur (UserService) et le repository (Repository) pour prendre en compte la gestion des images. Lors de l'ajout ou de la mise à jour d'un utilisateur, vous devez enregistrer le chemin de l'image dans la base de données.
   - Tester l'application : Vous devez tester votre application pour vous assurer que les images peuvent être ajoutées et affichées correctement pour chaque utilisateur.

apcpedagogie.com

4. Conseils

- Documentez les étapes que vous avez suivies pour ajouter la gestion des images, y compris les modifications apportées au modèle, à l'interface utilisateur et aux opérations CRUD.
- Assurez-vous de bien comprendre les bases de l'application de gestion d'utilisateurs que vous avez créée précédemment, car ce projet s'appuiera sur cette base.
- Prenez le temps de planifier la manière dont vous allez stocker les images (chemins de fichiers locaux ou URL distantes) et comment vous allez les afficher dans l'interface utilisateur.
- Si vous souhaitez relever un défi supplémentaire, explorez des fonctionnalités avancées telles que la capture d'image à partir de l'appareil photo, le stockage d'images dans le cloud ou l'optimisation de la taille des images.
- Cette description aidera vos apprenants à comprendre l'importance de maîtriser le projet précédent et à se préparer à ajouter des images à leur application de gestion d'utilisateurs. Ils devront tirer parti des compétences qu'ils ont acquises dans le projet précédent pour réussir celui-ci.

5. Etapes générales

1. **Stockez l'image de l'utilisateur :**
   - Assurez-vous d'avoir une manière de stocker l'image associée à chaque utilisateur.
   - Cela pourrait être le chemin du fichier de l'image dans le système de fichiers de l'appareil ou une URL vers une image distante.
   - Par exemple, vous pourriez ajouter un nouvel attribut à votre modèle User, tel que String? image.
2. **Chargez et affichez l'image :**
   - Dans l'interface utilisateur où vous affichez les détails de l'utilisateur (par exemple, dans ViewUsers.dart), utilisez un widget d'image (comme Image ou NetworkImage) pour charger et afficher l'image associée à chaque utilisateur. Utilisez la valeur de l'attribut image de l'objet User pour spécifier la source de l'image.
3. **Exemple d'utilisation du widget Image :**

apcpedagogie.com

```
Image.asset('chemin_vers_l_image_locale.jpg') // Pour les images locales
```

- Exemple d'utilisation du widget NetworkImage pour les images distantes (URL) :

```
Image.network('https://exemple.com/chemin_vers_l_image_en_ligne.jpg')
```

4. **Stockez et gérez les images correctement :**

   - Si vous stockez les images localement sur l'appareil, assurez-vous de gérer leur chargement, leur stockage et leur suppression correctement. Vous devrez peut-être également gérer les permissions d'accès aux fichiers.
   - Si vous affichez des images distantes, assurez-vous que les URL des images sont correctes et que vous gérez les erreurs de chargement d'image de manière appropriée.

5. **Mettez à jour votre schéma de base de données :**

   - Mettez à jour votre schéma de base de données pour inclure la colonne "image" dans la table "users". Vous pouvez le faire en modifiant la méthode _createDatabase de la classe DatabaseConnection dans votre fichier database_connection.dart. Voici comment vous pouvez le faire :

```
Future _createDatabase(Database database, int version) async {
  String sql = '''
    CREATE TABLE users (
      id INTEGER PRIMARY KEY,
      name TEXT,
      contact TEXT,
      description TEXT,
      image TEXT  // Ajoutez cette ligne pour la colonne "image"
    );
  ''';
  await database.execute(sql);
}
```

   - Après avoir modifié le schéma de la base de données, assurez-vous que vous avez supprimé votre base de données existante (si elle existe déjà) pour que la nouvelle structure soit prise en compte. Vous pouvez également augmenter la version de la base de données pour forcer la création de la nouvelle structure.

apcpedagogie.com

6. **Mettez à jour le modèle User :**
    - Comme vous l'avez déjà fait, ajoutez un attribut String? image à votre modèle User pour stocker le chemin de l'image associée à chaque utilisateur.
7. **Mettez à jour les opérations CRUD :**
    - Si vous permettez aux utilisateurs d'ajouter ou de modifier des images, assurez-vous que vos opérations CRUD (dans UserService et Repository) prennent également en compte la gestion des images. Vous devrez enregistrer le chemin du fichier local ou l'URL de l'image dans la base de données lorsque vous ajoutez ou mettez à jour un utilisateur.
    - Modifiez les opérations CRUD (dans UserService et Repository) pour prendre en compte la gestion des images. Lors de l'ajout ou de la mise à jour d'un utilisateur, enregistrez le chemin de l'image dans la base de données.
8. **Affichez et manipulez les images dans l'interface utilisateur :**
    - Assurez-vous que vos écrans (AddUser.dart, EditUser.dart, etc.) permettent aux utilisateurs de sélectionner ou de télécharger des images.
    - Vous devrez également afficher les images actuelles des utilisateurs dans l'interface utilisateur lorsque vous affichez leurs détails.
    - Modifiez les écrans d'interface utilisateur (AddUser.dart et EditUser.dart) pour permettre aux utilisateurs de sélectionner ou de télécharger des images et enregistrez le chemin de l'image dans l'objet User.
9. **Affichez les images dans les listes :**
    - Modifiez l'écran ViewUsers.dart (ou tout autre écran de liste) pour afficher les images associées aux utilisateurs dans la liste.
    - Utilisez un widget d'image (comme Image ou NetworkImage) pour charger et afficher l'image associée à chaque utilisateur.
    - Utilisez la valeur de l'attribut image de l'objet User comme source de l'image.

Cours Flutter

Source : https://apcpedagogie.com/projet-flutter-application-de-gestion-des-utilisateurs-crud-2 · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com