



Création de EDVIGE INSA

Après le fichier EDVIGE qui a tant fait parler de lui, le département Informatique est le premier à mettre en place EDVIGE-INSA, un fichier recensant tous les utilisateurs du département. Vous avez pour but de développer une version mobile de cette application.

REMARQUE : LA VERSION NON MOBILE CONTENANT DES DONNEES CONFIDENTIELLES, NE PEUT VOUS ETRE COMMUNIQUEE.

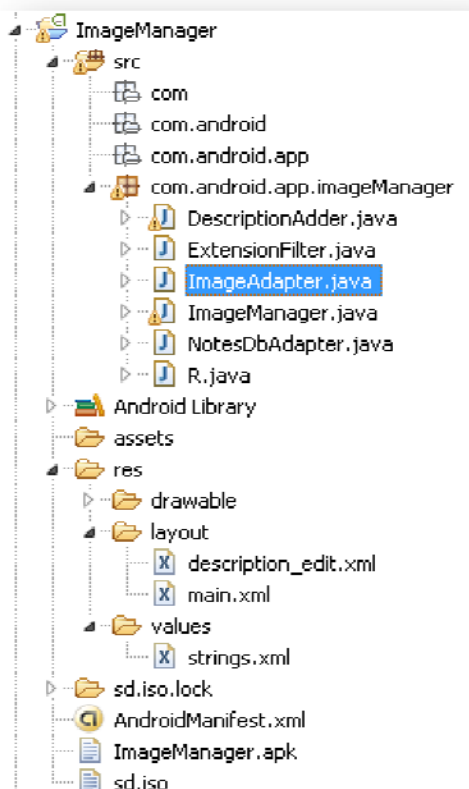
À travers le développement de cette application, vous découvrirez certains aspects de la programmation sous Android.

Mise en place de la plateforme de développement

Vous pouvez récupérer l'ensemble du TP et son environnement (eclipse et SDK de Android) à l'adresse indiquée au tableau et extraire l'archive dans le répertoire D:\Tmp.

Vous pouvez maintenant lancer la version d'eclipse située dans le répertoire extrait. Vous avez devant vous un squelette du projet à réaliser.

LANCEZ L'APPLICATION DES MAINTENANT CAR LE CHARGEMENT DE L'EMULATEUR EST LONG.



Voici une description des fichiers que l'on va utiliser.

JAVA

ImageManager.java :

C'est l'activité principale de l'application lancée par Android.

- **DescriptionAdder.java :**

C'est l'activité secondaire qui permet d'ajouter une description à une image

- **ImageAdapter.java :**

C'est la classe qui permet de convertir des images en Bitmap, seul format affichable nativement

- **R.java :**

Fichier auto-généré à n'ABSOLUMENT pas modifier

- **NotesDbAdapter.java :**

L'interface entre la base de données présente dans le téléphone et l'application

(Aucune modification nécessaire)

- **ExtensionFilter.java :**

Implémentation d'un filtre simple basé sur l'extension d'un fichier *(Aucune modification nécessaire)*

XML

- **main.xml** :

Ce fichier décrit l'interface graphique de l'activité ImageManager.

- **description_edit.xml** :

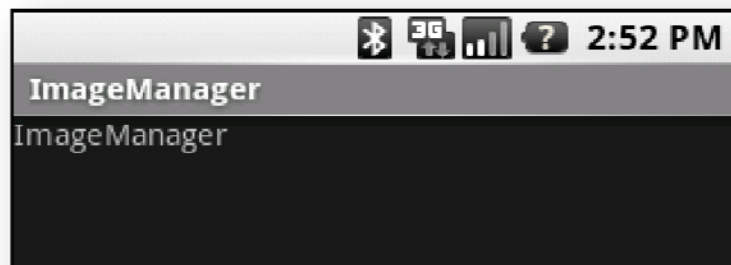
Ce fichier décrit l'interface graphique de l'activité DescriptionAdder

- **strings.xml** :

Ce fichier recense des chaînes de caractères pouvant apparaître dans le projet.

L'EMULATEUR DEVRAIT ETRE LANCE (ET ANDROID DEMARRE) DEPUIS LE TEMPS...

Votre affichage devrait ressembler à ça :



C'est déjà très joli, mais pour l'instant, l'utilité reste à prouver...

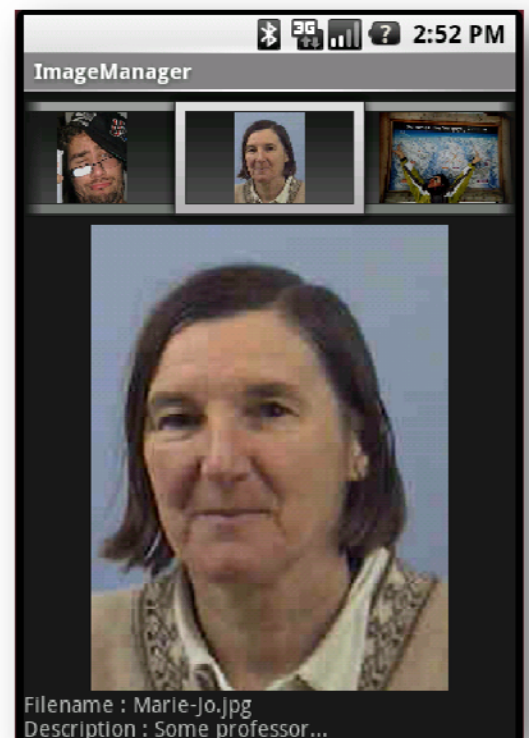
Réalisation de la visionneuse d'images

1. Interface Graphique

Pour commencer, il faut *designer* l'interface graphique de l'activité principale. Pour cela, il faut modifier le fichier *main.xml* en y ajoutant une liste d'images (*Gallery*), une image centrale (*ImageView*) et une zone de texte (*TextView*). N'oubliez pas d'utiliser un *Layout* qui contient tous vos éléments.

Cherchez à obtenir un *design* proche de celui-ci. Notez que l'éditeur graphique des *layouts* n'est pour le moment pas fonctionnel. Il faudra donc tout faire à l'ancienne, en codant. Pour tester votre *layout*, il faudra donc exécuter le projet.

En effet, quand vous cliquez sur le bouton d'exécution, eclipse génère un fichier *.apk* qui correspond à une application Android. Ensuite, il l'installe dans l'émulateur comme on installe un programme sur Windows. Bien entendu, il remplace toute version précédente.

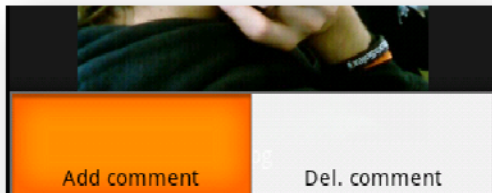




2. Evènements

Maintenant que l'interface est réalisée, on va s'attaquer à la communication entre ces objets. Pour cela, il faut gérer un évènement qui permet de changer l'image centrale et sa description (son nom de fichier) en fonction l'image sélectionnée dans la galerie.

3. Menu



Une fois que vous arrivez à visualiser tous les individus présents sur la carte SD, n'hésitez pas à rajouter un menu qui nous servira par la suite à noter des informations confidentielles sur ces personnes. Pour cela, il suffit d'implémenter des fonctions héritées de la classe *Activity* (voir liens plus bas)

4. Nouvelle activité

Nous allons maintenant nous intéresser à la façon dont communiquent deux Activités entre elles. Cela tombe bien car nous n'avons aucun outil pour préciser d'éventuels actes ou observations sur nos protagonistes. Il vous faut donc créer une Activité qui sera exécutée lors du clic sur un bouton du menu.

ATTENTION : NE CREEZ PAS DE NOUVELLE CLASSE, CELLE-CI EXISTE DEJA : ELLE A LE DOUX NOM DE DESCRIPTIONADDER

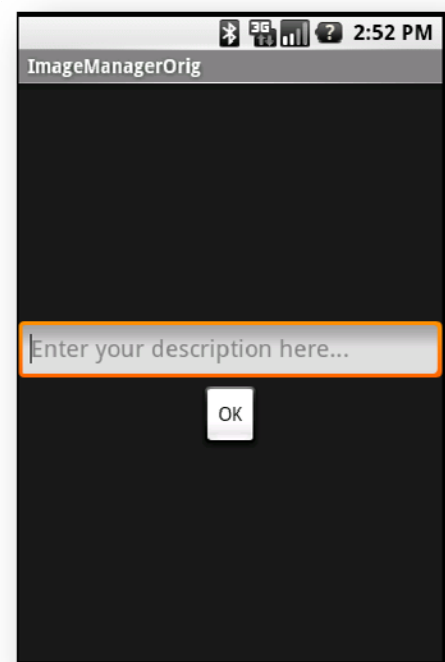
Elle est composée d'un *TextView* et d'un *Button* (encore une fois, vous avez le droit d'exprimer votre talent de designer). Lorsqu'une description est ajoutée, l'activité ajoute une entrée dans une base de données que nous avons créée.

Cette base de données consiste en une table composée de 2 colonnes : *filename* et *description*. À terme, le but est de conserver les descriptions d'une session sur l'autre. En effet, pour ficher une personne, c'est préférable.

Dans la méthode *onCreate*, il faut gérer 2 choses :

- L'évènement associé au bouton de validation (notamment pour quitter l'activité et mettre à jour la base de données).
- La gestion du paramètre envoyé par l'activité mère (le nom de fichier).

Replaçons-nous maintenant du côté de notre activité principale. Il faut mettre à jour la description en récupérant les données à partir de la base fraîchement mise à jour.





Liens utiles

Pour tout savoir sur les activités d'android.

<http://code.google.com/android/reference/android/app/Activity.html>

Réaliser les tâches les plus communs sur android n'aura plus de secret pour vous.

<http://code.google.com/android/kb/commontasks.html>

Informations complémentaires

La classe NotesDbAdapter :

- La fonction **open()** permet d'ouvrir une connexion avec la base de données.
- La fonction **close()** permet de fermer cette connexion.
- La fonction **createOrUpdateNote(String filename, String description)** permet de créer (ou mettre à jour le cas échéant) une description associée au filename considéré.
- La fonction **deleteNote(String filename)** permet de détruire la description associée au filename.

