Application de produits et services locaux - LocaLim

Version du 11 octobre 2020, 20:45

Objectifs

Le but de ce projet est de réaliser une application permettant de stocker et d'échanger des informations sur des produits ou des services offerts localement et ce, de manières collaborative. Dans ce projet vous serez ammené à mobiliser les différentes notions abordées en cours.

1 Condition d'évaluation

Quelques indications :

- Le projet devra se réaliser en binôme.
- Vous devrez créer un repository Gitlab **privé** pour réaliser ce projet et inviter votre enseignant comme membre du projet. Ce passage par Git est obligatoire et sera pris en compte dans le suivi de votre avancé dans le projet.
- A la fin du projet (la date exacte vous sera donnée ultérieurement), vous devrez remettre sur la plateforme Community un zip de votre code source ainsi qu'un apk fonctionnel.
- Vous accompagnerez le code source d'un rapport de maximum 3 pages, présentant les principales fonctionnalités implémentées et détaillant votre environnement de test (en particulier les caractéristiques de votre émulateur, les logins/mdp pour pouvoir tester etc.).
- une séance de démonstration (10min par groupe) sera positionnée dans la mesure du possible pour que vous puissiez montrer l'ergonomie et les fonctionnalités de votre application.

Bien entendu, la qualité du code et l'ergonomie générale de l'application seront pris en considération dans l'évaluation.

2 Description générale du projet

L'application à réaliser est une application de prise de notes permettant de gérer à la fois des images et du texte.

▷ **Exercice 1** *Classes du domaine*
Vous devrez manipuler deux types d'objets, les `Offres` et une `ListeDOffres`

Une `Offre` se caractérise par un titre (`String`), un texte (`String`) présentant le produit ou service et une image (`Bitmap`. Il y aura également un lieu précisant l'endroit ou le service/produit est disponible.

Une `ListeDOffre` possèdera en attribut un `ArrayList` d'`Offre` et prendra en paramètre du constructeur un `Context`. En pratique on lui fournira l'activité qui l'a construite pour permettre à la `ListeDOffre` d'accéder aux différentes ressources de l'application.

▷ **Exercice 2** *Mise en place de la structure générale*
Pour démarrer, voici quelques indications sur l'ergonomie minimale à obtenir : Vous devez avoir, au minimum, trois activités :

- L'activité principale qui affiche la `ListeDesOffres`
- Une activité pour afficher le contenu d'une `Offre`
- Une activité pour saisir une nouvelle `Offre` (que vous écrirez dans l'exercice 3)

Mais en plus :

- La `ListeDOffre` doit se présenter sous forme de liste verticale.
- Une `Offre` se présente verticalement avec son titre en haut, l'image en dessous, le texte et la position en bas de l'écran. Elle doit de plus être scrollable pour permettre de voir tout le texte correctement sur un petit écran. De manière facultative, la position peut être affiché dans une carte.

Une version qui sera valorisée dans la notation, consiste à utiliser des fragments pour permettre l'adaptation de votre application à l'orientation de l'écran.

▷ **Exercice 3** *Premiere* `Offre`
Une fois la structure construite, il vous est recommandé de tester en spécifiant un titre et un texte dans le fichier et en mettant une image dans `drawable`. **Attention l'image doit être au format BMP.** Inutile de prendre une grosse résolution d'image, l'affichage sur les émulateurs ne supporte pas du 1080p ou du 4k...

Pour vous aider à charger l'image vous pouvez vous appuyer sur la classe `BitmapFactory` qui offre une méthode `decodeResource` permetttant d'obtenir un objet de type `BitMap`. L'affichage d'une Bitmap en Android est relativement simple et peut se faire à l'aide d'un layout `ImageView`.

Dans ce premier test vous devez donc précharger une offre, la stocker dans la liste, afficher la liste contenant un élément et l'afficher lorsque l'on clique dessus.

▷ **Exercice 4** *Ecriture d'une Offre*
Une fois que l'affichage d'une `Offre` fonctionne, passez à l'écriture d'une nouvelle `Offre`. Pour cela, vous pouvez dans un premier temps ajouter un bouton dans l'activité permettant d'afficher la `ListeDeNotes`. Néanmoins, une version plus élégante consiste à ajouter cette fonction dans un bouton flottant. Regardez la documentation sur le site Android : `https://developer.android.com/guide/topics/ui/floating-action-button`.

Pour les autres fonctionnalités comme la recherche qui viendra plus tard par exemple,

vous pouvez passer par la barre d'action (ou app bar) :`https://developer.android.com/training/appbar`

Par défaut la barre d'action permet d'ajouter des entrées textuelles. Il est néanmoins possible de remplacer ce texte par une icone. Un jeu d'icone conforme au style Android est disponible à l'adresse `https://developer.android.com/design/downloads/index.html`

L'activité d'édition d'une nouvelle `Offre` devra permettre d'éditer le titre (une seule ligne), de choisir un fichier image à insérer, de saisir le texte associé (qui peut être sur plusieurs lignes) et de définir le lieu où se trouve l'offre. Un bouton situé en bas de l'activité permettra de valider la création de la `Note`.

Pour choisir le fichier image, vous pouvez utiliser un `ImageView` cliquable. Une propriété intéressante de ce widget est qu'il permet d'afficher une image temporaire en attendant d'avoir son image définitive.

Cela peut se faire avec le code présenté sur la figure donnée en exemple.

```
<ImageView
    android:id="@+id/EditImageIntox"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_gravity="center_horizontal"
    android:maxHeight="200dp"
    android:src="@drawable/placeholder"
    android:onClick="pickImage"/>
```

FIGURE 1 – Exemple d'ImageView contenant une image temporaire

Une image temporaire peut être trouvée simplement en cherchant le terme "placeholder" dans google image. Prenez en une libre de droit bien entendu.

Un clic sur l'image temporaire fera apparaître une boite de dialogue permettant de choisir son fichier.

▷ **Exercice 5** *Fonctionnalités complémentaires*

La version précédente est la version minimale à atteindre. Quelques améliorations (ce n'est pas limitatif) peuvent être ajoutées :

- Utilisez le swipe pour passer d'une `Offre` à une autre lorsqu'elles sont affichées en plein écran.
- Laissez à l'utilisateur la possibilité de prendre une photo au lieu de choisir une image déjà stockée sur le système.
- Ajoutez des cartes pour localiser des informations.
- Implémentez une fonction de recherche textuelle (pouvant supporter des tags qui seraient ajoutés aux offres par exemple).
- Ajoutez une fonction de recherche par position qui fait apparaitre sur la carte les

offres situées à proximité.
— N'hésitez pas à être créatifs !

▷ **Exercice 6** *Et le collaboratif alors ?*

La collaboration et le partage de notes se fera à travers la plateforme Firebase. En particulier vous devrez gérer :

— L'enregistrement et l'authentification des utilisateurs à travers la plateforme Firebase.
— Supporter la base de donnée temps réel pour permettre à plusieurs utilisateurs de synchroniser leurs offres de manière transparente.
— Utiliser le stockage cloud pour héberger les images liées aux notes et pouvoir ainsi les partager plus simplement.

Pensez à tester votre application avec deux émulateurs pour vérifier que la synchronisation fonctionne bien.