|  |  |
| --- | --- |
| **420-14D-FX Développement pour appareils mobiles**  **Matthieu Hermet Automne 2021** | **Travail Pratique #3** |

Objectif

Dans ce travail, vous devez réaliser une application Android de consultation/création de points d’intérêts sur le thème des activités plein-air.

Cette application est composée d’une activité principale servie sous trois fragments : une carte, une liste de points d’intérêt et une page de détail correspondante à chaque point d’intérêt.

L’utilisateur peut ajouter des points d’intérêts depuis la carte. Ces points sont alors stockés sous forme d’objets dans une base de données.

Au démarrage de l’application, les points d’intérêts stockés dans le BD sont affichés dans la liste sous forme de boîtes listées, ainsi que sur la carte sous forme de marqueur cliquables. En outre, le point de géolocalisation de l’utilisateur sert de point de positionnement dans la carte.

Une application Android préconstruite est fournie en complément de cet énoncé : **TP3\_Depart**. Les layouts sont déjà en place (hors versions paysage), ainsi que certains éléments de programmation, et il vous faut donc examiner le code minutieusement afin de comprendre quels ajouts sont nécessaires.

Voir en annexe la liste des fichiers à compléter.

Pondération

Ce travail pratique compte pour 25% de votre note finale.

Équipe

Ce travail est individuel.

Remise

Vous devez faire **une remise électronique** par LÉA. Les remises doivent se faire avant 23h59 le dimanche 14 novembre 2021.

Description de la remise

Ce travail pratique porte sur les aspects suivants du développement pour appareils mobiles :

* API Google Map et Location
* Recycler View
* View Model et LiveData
* View Binding
* Navigation
* Permissions

**L’application à développer comporte une seule activité**. Cette activité permet l’accès à 2 fragments depuis un menu de navigation :

* Fragment Carte
* Fragment liste (Recycler View)

De plus, le fragment liste permet l’accès à un fragment lorsqu’on clique sur un élément de la liste :

* Fragment Détail

L’application manipule un type d’objet Point d’intérêt, constitué comme suit :

* Nom
* Catégorie
* Adresse civique
* Points de géocoordonnées

Ces objets sont stockés dans une BD de type SQLite (*Room*).

**Voir les spécifications ci-après pour les détails de l’implémentation**.

## Activité principale

L’activité principale n’a pas d’affichage propre. Au démarrage elle affiche par défaut le fragment liste.

En revanche elle déploie un menu de navigation de type *Drawer* dans une *ToolBar*.

Ce menu de navigation permet d’accéder à 2 fragments : carte et liste.

Vous devez mettre en place l’affichage correct de ce menu dans cette activité.

# *Fragment Carte*

Ce fragment est accessible depuis le menu de navigation et affiche une carte (Google Map API) incluant un contrôleur de zoom, ainsi qu’un bouton de type *Floating Action Button*.

À l’affichage, la carte se place sur la position courante de l’utilisateur (Géolocalisation). À la première utilisation de l’app, une demande de permission doit être adressée à l’utilisateur (*Manifest.permission.ACCESS\_FINE\_LOCATION*).

Également, à l’affichage, la liste des points d’intérêts stockés dans la BD sont récupérés (via son *ViewModel*) et affichés sur la carte sous forme de marqueurs (*marker*). Ces marqueurs sont cliquables.

Le fab inclus dans le fragment active un mode *ajout de point d’intérêt* lorsqu’il est cliqué.

Sur cette carte. 2 comportements dynamiques :

1. Mode *ajout de point d’intérêt* : permet d’ajouter un point d’intérêt à la BD. Activé par clic sur le *fab*.

Lorsque ce mode est activé, l’utilisateur peut ajouter un point d’intérêt à la BD par simple clic sur la carte à l’emplacement propice. Le *fab* doit changer de couleur de sorte à indiquer de façon visible que ce mode est activé.

Au clic sur la carte :

* 1. un formulaire d’ajout comprenant 2 champs (*EditText* Nom et *Spinner* Catégorie) est affiché et permet la saisie d’une valeur de nom et d’une catégorie pour ce point d’intérêt.
  2. l’adresse civique pour ce point d’intérêt est obtenue depuis l’API Google Location. Ces éléments, auxquels viennent s’ajouter les valeurs de latitude et de longitude (reçus depuis *onMapClick*), permettent la construction d’un objet *point d’intérêt* en bonne et due forme à stocker dans la BD.

NOTE : les objets *points d’intérêts* manipulés dans ce fragment sont encapsulés dans un objet de type *LiveData* et servis depuis le *ViewModel* – ainsi, **dès la validation d’un ajout de point d’intérêt dans la BD depuis le formulaire, la carte sera automatiquement mise à jour et le nouveau point créé apparaîtra sur la carte en tant que marqueur et dans la liste en tant qu’élément** (fragment liste).

1. Clic sur un marqueur : lorsque l’utilisateur clique sur un marqueur :
   1. une fenêtre d’information apparaît.
   2. la distance depuis la position courante de l’utilisateur et le point cliqué est calculée (API Location) et affichée sur le fragment Carte.

La fenêtre d’information affichée est composée :

* 1. D’une image correspondante à la catégorie du point (partie gauche)
  2. Des 3 éléments d’information principaux composants un point d’intérêt : nom, catégorie, adresse civique.

Lorsque cette fenêtre d’information est affichée, un simple clic hors de la zone fenêtre sur la carte permet de revenir sur la carte. Attention à bien régler ce comportement, de sorte à ce qu’un clic sur la carte de sortie de fenêtre d’information marqueur n’entraîne pas l’affichage du formulaire d’ajout de point d’intérêt.

## Fragment liste des points d’intérêts

Ce fragment affiche une liste de type *Recycler View*. Il est affiché au démarrage de l’application et demeure accessible depuis le menu de navigation.

Au chargement du fragment, les points d’intérêts sont affichés sous forme d’éléments de liste.

Le règlement d’affichage des éléments est géré depuis un adapteur (*RecyclerView.Adapter*) chargé de produire un placement correct des propriétés composants un point d’intérêt dans le layout rangée.

La rangée décrivant un point d’intérêt est composée :

1. D’une image correspondante à la catégorie du point (partie gauche)
2. Des 3 éléments d’information principaux composants un point d’intérêt : nom, catégorie, adresse civique.

Sur cette liste, 1 comportement dynamique :

1. Affichage fenêtre Détail : une fenêtre de détail est affichée lorsque l’utilisateur clique sur une rangée de la liste.

Ce comportement est un effet de *navigation* réglé dans l’adapteur, au clic sur un élément de la liste.

**Cette fenêtre est un nouveau fragment**. Ce fragment n’est pas servi par le menu de navigation déroulant, mais il est inclus dans le schéma de navigation de sorte à régler le comportement du bouton Up automatiquement (retour vers fragment Liste).

Cette fenêtre affiche une vue du point d’intérêt sous forme détaillée. Elle est composée comme suit :

* 1. Une image (catégorie) de taille suffisante à occuper la largeur du conteneur parent (le fragment Détail).
  2. Les 3 éléments d’information principaux composants un point d’intérêt : nom, catégorie, adresse civique.
  3. Une carte (API Google Map) incluse dans le fragment et centrée sur le point d’intérêt détaillé (marqueur).

**NOTE : chaque affichage de points dans l’application (Marqueur, Éléments de liste, Vue Détail) comprend une image en plus des informations de points. Les images doivent varier selon les catégories.**

# Caractéristiques générales du code

* **Une attention particulière sera portée aux commentaires descriptifs de votre application, qui compteront pour 10% de la note.**

# Caractéristiques générales et techniques de l’app

Cette application Android doit avoir les aspects suivants :

* La clé d’API (Google Map) définie pour votre version ne doit pas avoir de restrictions. (Liste des clés dans la Console Google Cloud 🡪 Éditer 🡪 Aucune).
* L’application doit supporter la rotation (layouts).
* Vous organiserez la disposition des éléments et styler l’apparence générale des vues comme bon vous semble en respect de cet énoncé.
  + **Une partie mineure de l’évaluation sera consacrée à noter l’effort que vous aurez mis à personnaliser cette application.**
* L’application doit présenter un nom propre et une icône spécifique sur l’écran d’accueil.
  + L’icône doit se décliner en 4 densités : hdpi, mdpi, xhdpi et xxhdpi.

# Contexte d’évaluation

* Android API >= 17.
* SDK cible API 30 ou supérieur.
* L’application doit supporter les rotations d’écran sans problème.

# Contraintes

* Votre application doit être développée avec Android Studio et l’ADT.

À remettre

Vous devez remettre **un** document présentant les éléments suivants :

* Page de présentation d’un .doc contenant toute information que vous jugerez utile de fournir.
* Un zip de votre application.
* Une version distribuable de votre application.
  + Fichier .apk que vous trouverez dans le répertoire **app\build\outputs\apk** du répertoire de votre application.

Annexe : liste des fichiers à compléter

Classes :

* LocationDao
* DetailsFragment
* DetailsViewModel
* HomeFragment
* HomeViewModel
* MapsFragment
* MapsViewModel
* LocationAdapter
* MainActivity

Voir l’application pour le détail des éléments à ajouter.

Ressources :

* mobile\_navigation.xml

Doivent être définis les actions de navigation nécessaires ainsi que les arguments passés.