

DOCUMENTATION

Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert. [Sur 2 points]

PREUVE :

(Voir [ContexteDiagrammes.pdf](#))

Je sais faire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application. [Sur 5 points]

PREUVE :

(Voir [ContexteDiagrammes.pdf](#))

Je sais concevoir un diagramme UML de qualité représentant mon application. [Sur 7 points]

PREUVE :

(Voir [ContexteDiagrammes.pdf](#))

Je sais décrire un diagramme UML en mettant en valeur et en justifiant les éléments essentiels. [Sur 6 points]

PREUVE :

(Voir [ContexteDiagrammes.pdf](#))

PROGRAMMATION

Je sais utiliser les Intent pour faire communiquer deux activités. [Sur 1 point]

PREUVE :

Au début de l'application, nous utilisons des intents pour naviguer entre les différentes activités de notre mais nous avons changer ces activités par des fragments pour que notre application soit plus simple à maintenir.

Je sais développer en utilisant le SDK le plus bas possible. [Sur 1 point]

PREUVE :

Nous n'utilisons aucune fonctionnalité qui demande un SDK plus élevé que celui qui touche toutes les utilisatrices et tous les utilisateurs.

Je sais distinguer mes ressources en utilisant les qualifier. [Sur 1 point]

PREUVE :

- drawable
- drawable-hdpi
- drawable-mdpi
- drawable-v21
- drawable-v24
- drawable-xhdpi
- drawable-xxhdpi
- layout
- menu
- mipmap-anydpi-v26
- mipmap-hdpi
- mipmap-mdpi
- mipmap-xhdpi
- mipmap-xxhdpi
- mipmap-xxxhdpi
- values
- values-fr
- values-v21

Effectivement nous utilisons des qualifieurs pour différencier les ressources des appareils de tailles différentes et/ou de langues différentes.

Je sais modifier le manifeste de l'application en fonction de mes besoins. [Sur 1 point]

PREUVE :

```

5
6 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
7 <uses-permission android:name="android.permission.INTERNET" />
8 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
9 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
10

```

Nous savons rajouter ou modifier le contenu du manifeste pour ajouter des permissions ou ajouter des accès aux activités (pas de preuve image car nous n'avons qu'une seule activité mais au cours du développement nous en avons utilisés).

Je sais faire des vues xml en utilisant layouts et composants adéquats. [Sur 1 point]

PREUVE :

Nous utilisons un DrawerLayout pour afficher notre carte de points et des LinearLayout pour les fragments qui sont des listes de possibilités donc nous pensons que ce sont des layouts adaptés.

Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les évènements. [Sur 1 point]

PREUVE :

```

80 @Override
81 public void removePOI(String id){
82     Keyboard.close(this);
83     map.removePointOfInterestViaID(id);
84 }

```

Nos activités et nos fragments ne font que des requêtes à faire par le modèle.

Je sais coder une application en ayant un véritable métier. [Sur 2 points]

PREUVE :

```

Map.java
PointOfInterest.java
Type.java

```

Notre modèle Map sait utiliser nos classes métier PointOfInterest et Type.

Je sais parfaitement séparer vue et modèle. [Sur 1 point]

PREUVE :

```

model
util
view

```

Nous avons effectivement une partie vue et une partie modèle où le modèle est appelé dans la vue et qui appelle le métier car la vue ne doit pas agir directement sur le métier.

Je maîtrise le cycle de vie de mon application. [Sur 1 point]

PREUVE :

Nous savons maîtriser le cycle de vie de notre application puisque nous gérons la rotation de l'écran ainsi que la fermeture de l'application.

Je sais utiliser le findViewById à bon escient. [Sur 1 point]

PREUVE :

```
56 //Drawer
57 DrawerLayout drawer = findViewById(R.id.drawer_layout);
58 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
```

Utilisation d'une variable pour stocker le résultat du findViewById, gourmand en ressources par il doit lire le fichier où sont stockés les Ids.

Je sais gérer les permissions dynamiques de mon application. [Sur 1 point]

PREUVE :

Nous n'avons pas de permissions dynamiques dans notre application.

Je sais gérer la persistance légère de mon application. [Sur 1 point]

PREUVE :

```
31 public static void saveMap(Map map, Context context){
32     SharedPreferences prefs = context.getSharedPreferences(MAP, Context.MODE_PRIVATE);
33     @SuppressWarnings("CommitPrefEdits") SharedPreferences.Editor editor = prefs.edit();
```

Nous savons utiliser des SharedPreferences afin de sauvegarder, ici notre carte de points, pour gérer par exemple les fermetures de l'application ou la rotation de l'écran.

Je sais gérer la persistance profonde de mon application. [Sur 1 point]

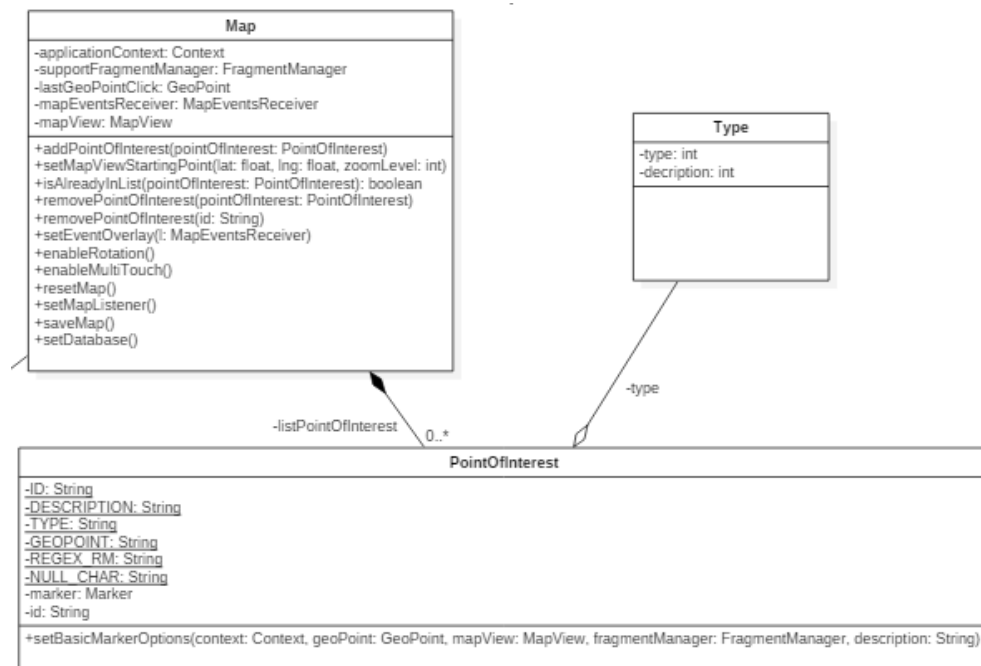
PREUVE :

```
162 final POIFirebase tmp = new POIFirebase(pointOfInterest.getId(), pointOfInterest.getLat(),
163     DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child(tmp.id);
164     reference.addListenerForSingleValueEvent(new ValueEventListener() {
```

Nous utilisons une base de données Firebase afin de sauvegarder notre carte de points pour qu'elle soit synchroniser sur tous les appareils qui utilisent l'application.

Je sais afficher une collection de données. [Sur 1 point]

PREUVE :



Nous savons effectivement afficher une collection puisque notre carte (Map) contient une liste de points d'intérêt (PointOfInterest).

Je sais coder mon propre adaptateur. [Sur 2 points]

PREUVE :

Nous n'avons pas d'adaptateur dans notre code.

Je maîtrise l'usage des fragments. [Sur 2 points]

PREUVE :

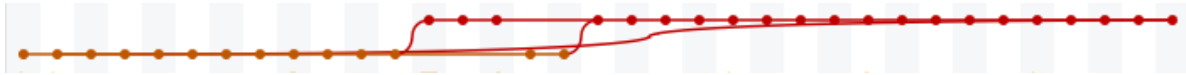
```

119         if (id == R.id.nav_settings) {
120             |         openFragment(new SettingsFragment());
121         }
122         else if (id == R.id.nav_about) {
123             |         openFragment(new AboutFragment());
124         }
    
```

Nous savons utiliser les fragments pour les ouvrir ou les fermer afin d'avoir une application dynamique.

Je maîtrise l'utilisation de Git. [Sur 1 point]

PREUVE :



Création de branches pour travailler sur des fonctionnalités sans interférer sur celle de l'autre.

APPLICATION

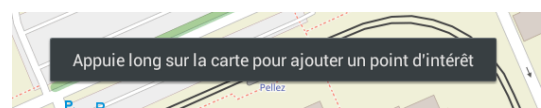
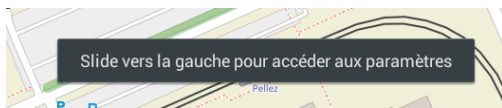
Mon application présente un intérêt à être publié sur le store. [Sur 5 points]

PREUVE :

Notre application a un but : donner la chance aux utilisateurs de pouvoir s'exprimer pour qu'ils puissent partager leurs lieux préférés avec l'ensemble de la communauté des utilisateurs.

Mon application fonctionne de manière à être utilisée par le public. [Sur 5 points]

PREUVE :



L'application est conçue pour que le public sache l'utiliser grâce à des conseils au lancement de l'application.

Mon application utilise des contraintes spécifiées lors du choix du projet. [Sur 5 points]

PREUVE :

```
162 final POIFirebase tmp = new POIFirebase(pointOfInterest.getId(), pointOfInterest.getLat(),  
163 DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child(tmp.id);  
164 reference.addListenerForSingleValueEvent(new ValueEventListener() {
```

Comme dit précédemment, nous utilisons une base de données Firebase afin d'avoir accès à notre carte de points sur tous les terminaux équipés de l'application.

Mon application utilise les contraintes à bon escient. [Sur 5 points]

PREUVE :

Cette base de données Firebase a une réelle utilité puisqu'elle permet de partager un point d'intérêt sur la carte avec l'ensemble des utilisateurs.