

Contexte

L'application VanLife est une application qui permet de localiser des points d'intérêt sur une carte. Un point d'intérêt est constitué d'un type et d'une description.

Le type peut être un parking, une douche, un point de vue... Le type est matérialisé par un symbole sur la carte.

La description est un texte court décrivant précisément le point d'intérêt (exemple : parking couvert sous le centre commercial).

On peut créer un point d'intérêt en faisant un appui long sur la carte : le point sera alors créé sous le doigt de l'utilisateur.

On peut facilement consulter un point d'intérêt en cliquant sur le symbole correspondant et ainsi accéder à la fiche ouverte du point d'intérêt. En accédant à un point d'intérêt, on peut le modifier et même le supprimer.

Diagramme de cas d'utilisation

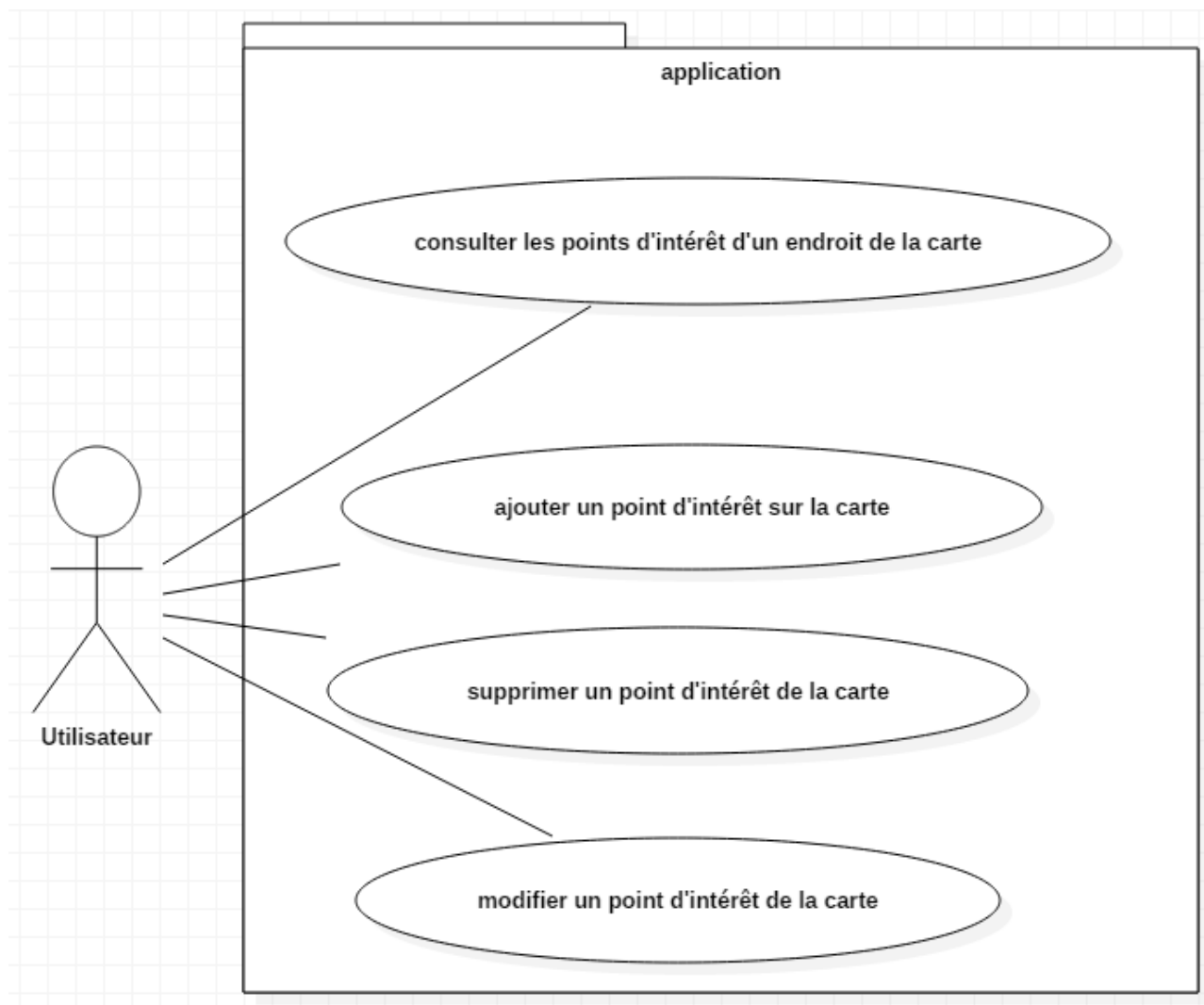
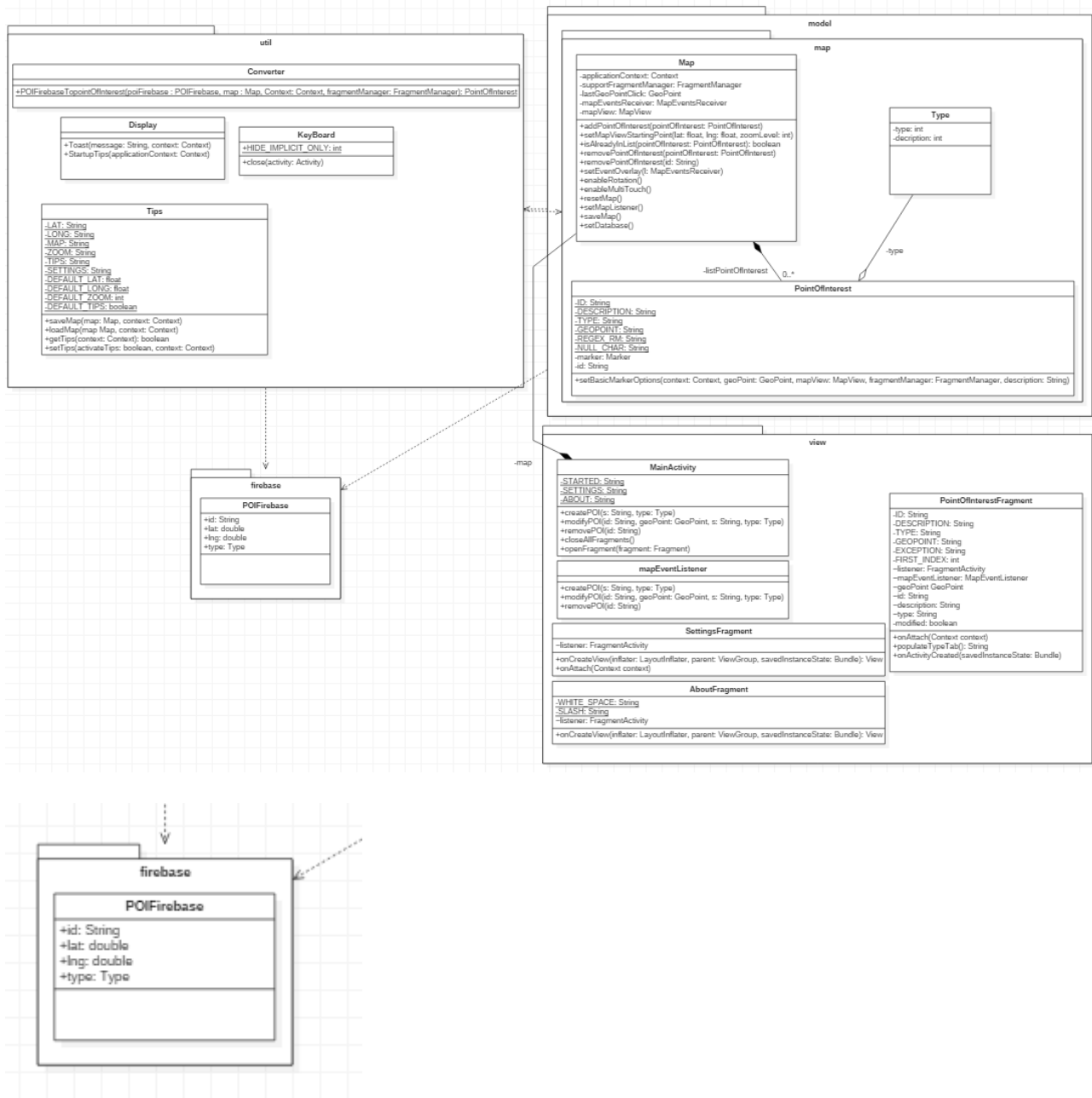
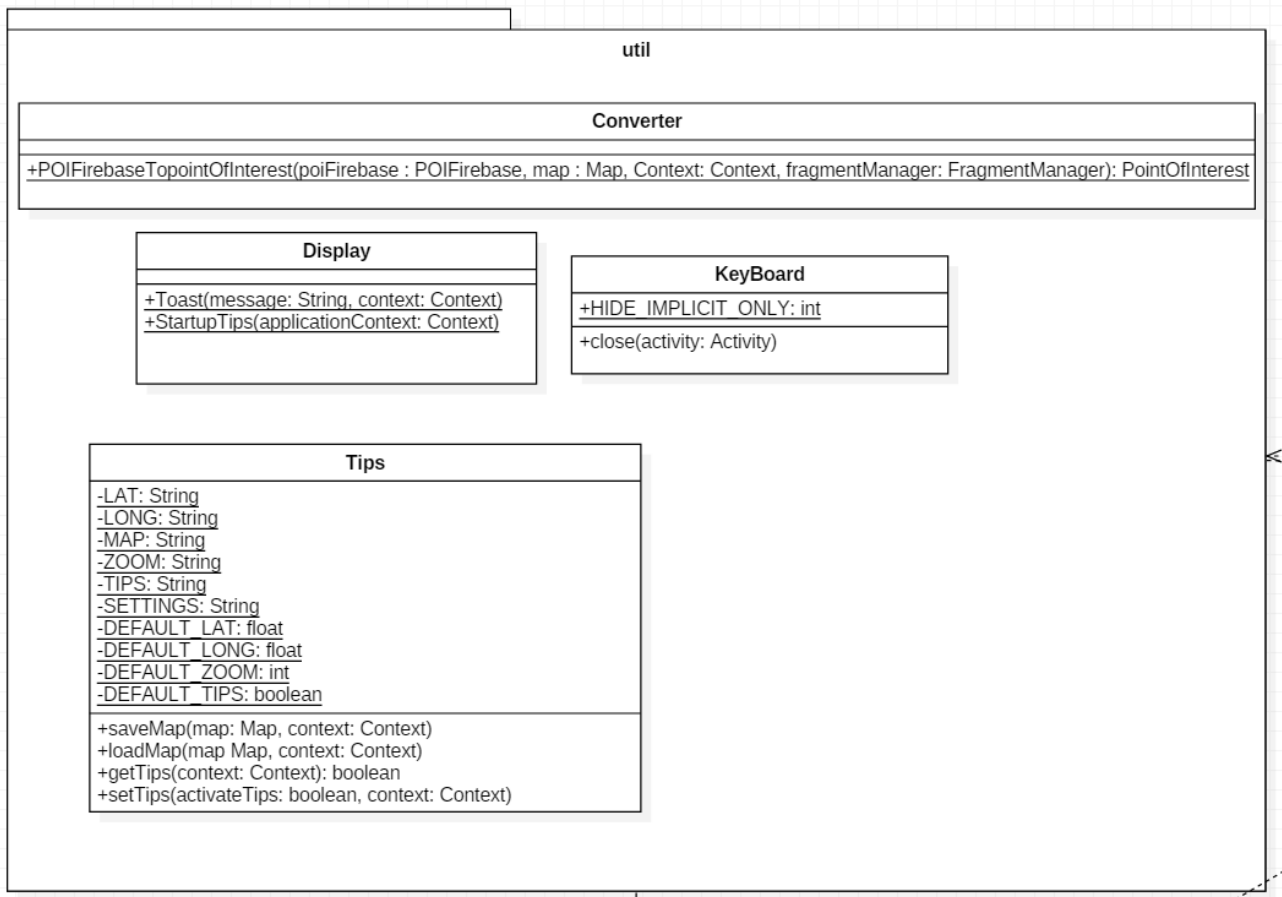


Diagramme de classes



La classe POIFirebase est la classe qui représente un point d'intérêt sur la base de données Firebase



Ce package représente toutes les classes dont on aura besoin pour convertir ou afficher quelque chose

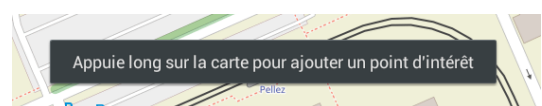
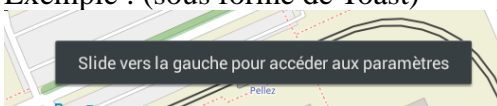
La classe Converter est une classe qui peut convertir un POI (PointOfInterest) en POIFirebase grâce à une méthode statique

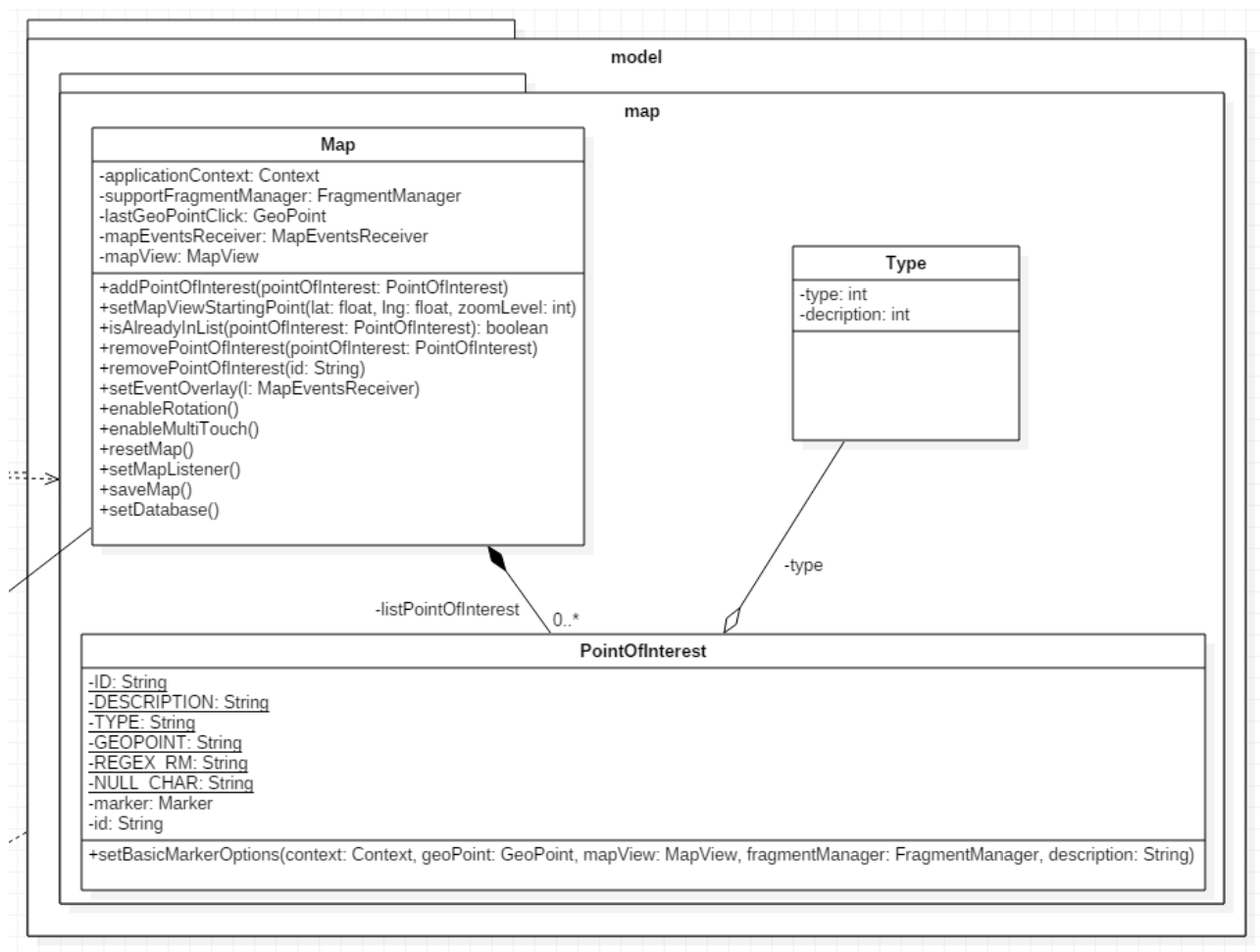
La classe Display est une classe qui permet d'afficher des informations sous forme de Toast en bas de l'écran grâce à une méthode de classe et une autre méthode de classe qui affiche les conseils de départ pour les nouveaux utilisateurs.

La classe KeyBoard représente le clavier et a une méthode qui ferme ce clavier (en cas de fin de saisie)

La classe Tips représente un conseil pour les nouveaux utilisateurs

Exemple : (sous forme de Toast)



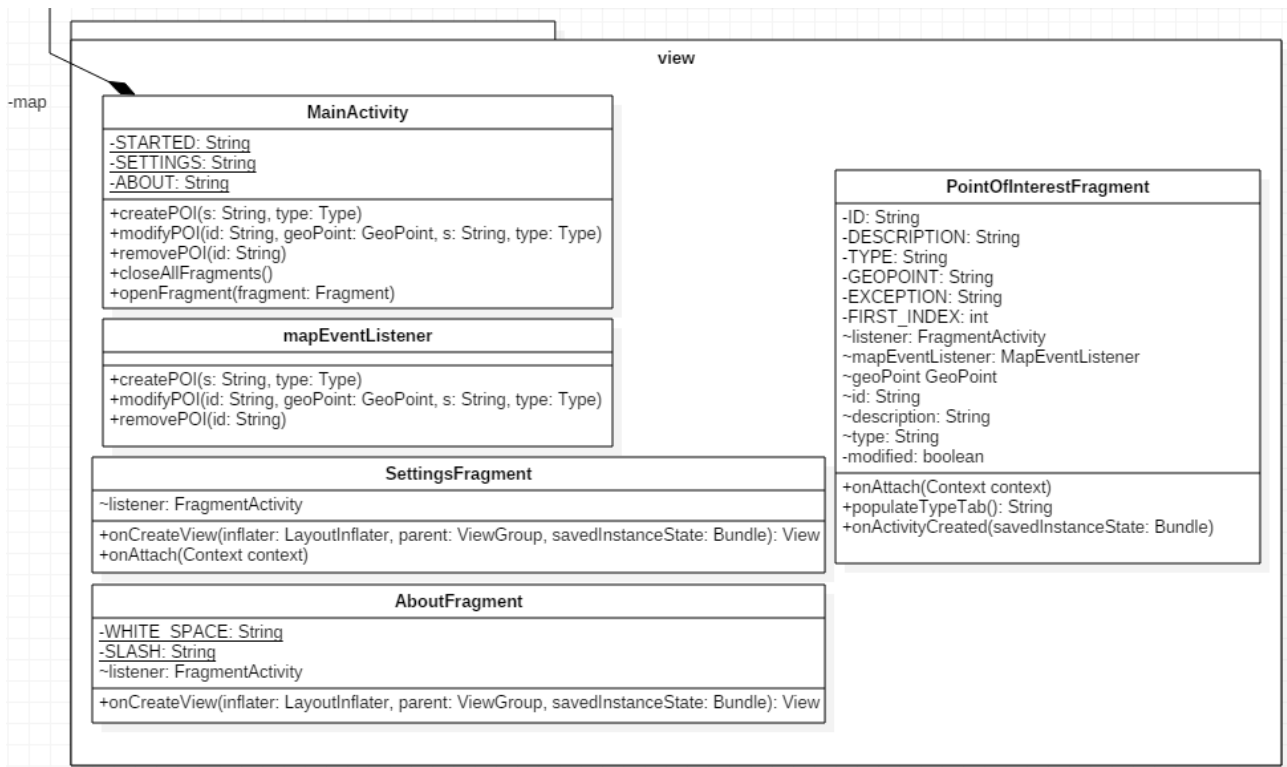


Ce package représente le modèle de notre application

La classe Map est une classe qui représente une carte qui contient tous les points d'intérêt. La classe gère les interactions des activités/fragments et des PointOfInterest et s'occupe donc d'ajouter, de modifier ou de supprimer les points d'intérêt

La classe PointOfInterest représente un point d'intérêt. Elle possède un type et un emplacement sur la carte.

L'énumération Type représente le type d'un point d'intérêt avec une image et une description. Elle regroupe tous les types de points d'intérêt possibles



L'interface `mapEventListener` possède la liste des méthodes à implémenter pour gérer une carte de points d'intérêt (Map).

La classe `MainActivity` représente le code qui fait le lien entre la vue et le modèle. Elle est liée à la vue principale.

La classe `PointOfInterestFragment` représente le code derrière le fragment `PointOfInterest`. Elle sert à gérer les interactions du fragment `PointOfInterest`.

Les classes `SettingsFragment` et `AboutFragment` représentent le code derrière les fragments `Settings` et `About`. Ces deux fragments servent respectivement à modifier des paramètres de l'application et à décrire l'application.