Améliorations possibles:

* Sauvegarder certaines préférences en lien avec le Profile d’un usager au niveau de la BDD afin de permettre leur synchronisation lorsqu’un même usager utilise de multiples appareils
* Utiliser un « Naviguation Drawer » (voir <https://developer.android.com/training/implementing-navigation/nav-drawer>) pour libérer l’espace dans la « Action Bar » dans le haut de la carte (ne laisser que « Search »)
* Un « Splash Screen » pour contrôler le flot d’initiation de l’état de l’application (synchronisation des informations avec la BDD, principalement) ainsi que pour rendre le code plus aéré (maintenance facilitée)
* Stocker les dates en tant que LONG dans la BDD (pour éviter les problèmes de TimeZone, et un ‘long’ prends moins d’espace qu’une String, etc.)
* Intégrer une vérification de la version de l’application afin de pouvoir gérer correctement les synchronisations (si une nouvelle version change quelque chose qui pourrait causer un crash dans une ancienne version, par exemple…)
* Conserver une StrongReference des NetworkRequests pour ne pas que le GarbageCollector les détruise
* Bug : « back to Home » qui fait en sorte que les Overlay de la Map ne s’affichent plus
* Réduire temps de traitement des opérations par le serveur (peut même causer des erreurs lorsque trop long)
* Features de recherche sur les alertes historiques (WebApp, surtout). Peut-être analyse IA ?
* Terminer de permettre l’entrée manuelle d’un nombre de mètres pour l’ajout d’une zone circulaire
* Importance d’une bonne plateforme de communication (code-lint, renvoie vers des messages spécifiques, etc.)

Min SDK = 16

Sharing ideas (aucunement des demandes, plus de la discussion... c'est le genre de chose qu'on pourrait mettre dans "Améliorations futures possibles" pour le rapport final):  
- DELETE, POST et PATCH Requests devraient toutes être authentifiées (entre autres pour des raisons de sécurité)  
- GET peut fort probablement ne pas devoir être authentifié (si on veut que des thrid-party puissent utiliser notre API, par exemple), mais ça pourrait être une brèche de [vie](https://www.facebook.com/messages/t/1493147314146533?fb_dtsg_ag=Ady83ldKGDD9BfbijHnynsXk3a97mF2O6PYAPAxzB9I7uQ%3AAdxkxPdxIkzqmNmU4TzIatbrlWuj85osFxj_srDID06egA) privée puisque nos instances de User possèdent de l'information sensible sur ces users (dont on n'a pas vraiment de besoin, d'ailleurs). On pourrait faire en sorte que les GET User nécessitent d'être authentifiés. La brèche est encore plus grande avec "GET All Users" puisque un SpamNetwork pourrait extraire les emails de tous nos usagers et les spammers, par exemple.  
- Les "Users" devraient avoir un champ qui indiquent leurs Préférences pour qu'on puisse sauvegarder/synchroniser/récupérer ces préférences pour plusieurs appareils

Pour un cas hypothétique comme le mode "Alertes GPS" qui utiliserait la localisation de l'usager pour faire des requêtes au serveur pour savoir s'ils y a des alertes autour de l'usager actuellement. dans la section "Préférences" de l'application mobile, l'usager pourrait sélectionner la fréquence de requête et/ou une distance minimale à avoir parcouru avant de faire une requête

Ou encore le simple fait qu'un usager se fou peut-être complètement des Alertes Usagers et alros il les a filtrées sur un de ses appareils, mais lorsqu'il se connecte sur un autre, cette Préférence n'aura pas été répliquée