

# HMIN309 - Programmation Cross-Platform Avec Xamarin.Forms

---

## TP Noté - Conception d'une application Cross-plateforme

Lors du cours sur l'IoT et Android Wear®, vous avez développé une application pour smartwatch capable d'envoyer un message à un serveur distant. Le message envoie également les coordonnées GPS de la montre-connectée de laquelle il a été envoyé.  
<https://github.com/greenpanther93/iot-basic-server>

Dans ce TP, nous allons concevoir une application qui va afficher la liste des messages envoyés au serveur. Cette application mobile sera réalisée à l'aide de Xamarin Forms et devra fonctionner sur les plateformes iOS et/ou Android (suivant les moyens à votre disposition, voir **indications**).

Par groupe de 2 ou 3 personnes maximum, vous allez implémenter dans votre application le cahier des charges ci-dessous.

Vous aurez à choisir entre l'implémentation de l'affichage par carte (choix 1), ou celui de la vue détaillée (choix 2). Si vous faites les deux, c'est encore mieux !

**Ce TP est à présenter pour mi-décembre.**

### Cahier des charges

---

• L'application doit être capable de se connecter au serveur RESTFUL afin de recevoir les messages envoyés par les montres-connectées. Cette application n'envoie pas de message au serveur (cette fonction est uniquement réservée aux montre-connectées). Elle comprendra trois écrans : l'écran d'**affichage par liste**, l'écran d'**affichage par carte** et l'écran **des détails**.

#### • L'affichage par liste

- Cet écran présentera l'ensemble des messages stockés sur le serveur. Il s'agit d'une liste vertical défilante, de façon à pouvoir parcourir l'ensemble des messages. Afin de rafraîchir la liste, l'utilisateur pourra cliquer un bouton « rafraîchir » **ou** attendre 30 secondes pour faire une mise à jour automatique.
- Les cellules de la liste devront afficher chacune un message. Le design des cellules est libre mais devront comporter au moins : le nom de l'émetteur et le début du message (disons les 50 premiers caractères).

#### • L'affichage par carte (choix 1)

- L'affichage par carte utilisera les coordonnées GPS des messages pour afficher tous les messages sur une carte du monde.
- Au-dessus de chaque point de la carte devront s'afficher les messages reçus.
- La carte devra être mise à jour en suivant le principe que l'affichage que par liste.

#### • L'écran des détails (choix 2)

- L'écran des détails s'active lorsque l'utilisateur clique sur un message. Il permet d'afficher le message en entier, mais aussi de voir tous les autres messages de l'utilisateur (s'il y en a).
- Les messages étant volatiles (ils peuvent disparaître après un rafraîchissement de la liste) on demandera d'ajouter une fonctionnalité de sauvegarde des messages dit *favoris*. Ces messages seront sauvegardés de façon permanente tant qu'ils seront marqués comme *favoris*. Les messages favoris s'afficheront toujours en premier dans la l'affichage par liste.

Ils apparaitront aussi avec une icône spéciale dans les autres vues afin dès les distinguer des autres messages non favoris.

- On souhaite aussi donner à l'utilisateur la possibilité d'associer à un émetteur une couleur particulière. Ainsi lors de l'affiche par liste ou par carte, l'utilisateur pourra voir apparaitre d'une couleur spécifique, tous les messages provenances de l'émetteur X.

## Indications :

---

- **L'implémentation**

1. Le design de l'application est libre mais devra correspondre au cahier des charges.
2. Vous devez implémentez le plus de points du cahier des charges possible !
3. La mise à jour des messages doit être fluide pour l'utilisateur : elle ne doit pas l'empêcher d'accéder au reste de l'application.

- **Déroulement de l'évaluation :**

1. L'évaluation prendra la forme d'une démo de 10 min.
2. L'évaluation est là pour vous permettre de présenter votre travail, de détailler vos choix d'implémentations, et de justifier les points que vous n'avez pu implémenter.
3. Votre projet doit être accessible en ligne. Utilisez par exemple votre compte **gitlab** de l'université : <https://gitlab.info-ufr.univ-montp2.fr> (identifiant et mot de passe identique à l'ENT)

Votre projet sera aussi téléchargé sur l'ordinateur de l'évaluateur qui tentera d'exécuter l'application pour iOS, Android ou Windows.