

Unicorn

DENNOUN Mohamed
PATIN Sandrine
PELLUET Sophie
ROLLAND-MONNET Cyril
SALOMON Arnaud



Mohamedennoun@gmail.com

Contexte

Le sujet est le bilan sur une application iOS, elle Permet d'obtenir des informations météorologique sur 8 jours mais l'endroit n'est pas indiqué

Les données sont mis à jour toutes les 3 heures

Notre rôle est d'analyser le code et de faire un listing des bugs ou incohérence lié au code

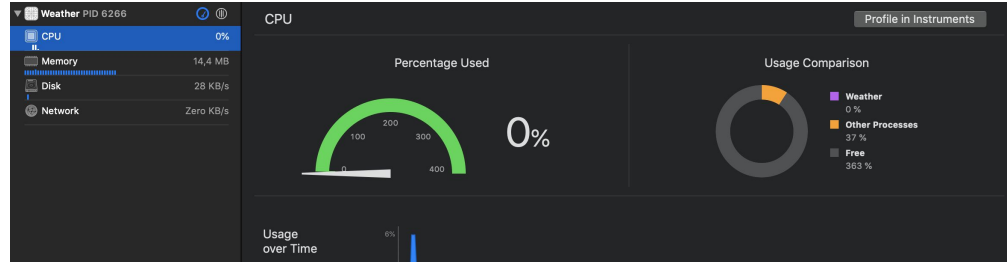


Architecture

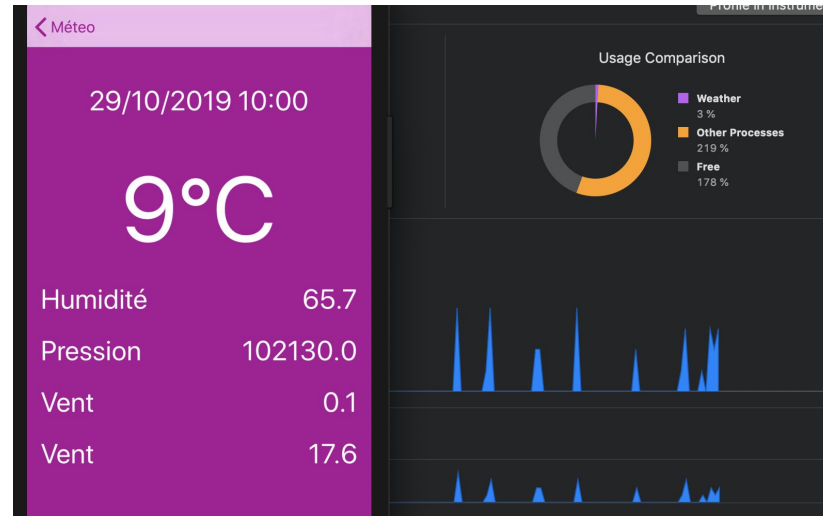
L'application va chercher ses données dans l'API de www.infoclimat.fr

L'application est recommandée pour une version d'iOS 8. le minimum recommandé est 11.0 cependant cette version obsolète car aujourd'hui, les appareils tournant sous iOS 13.

L'application

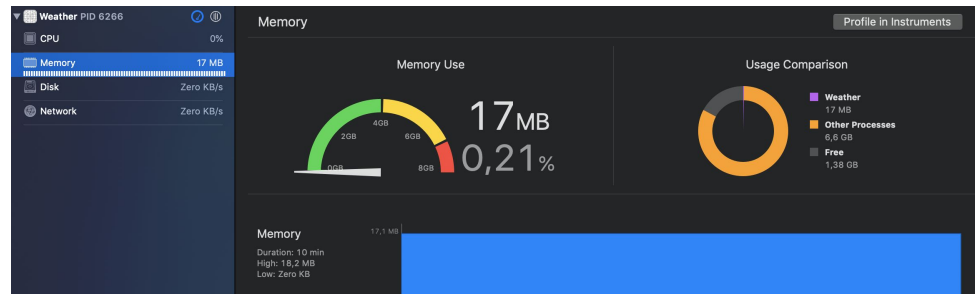
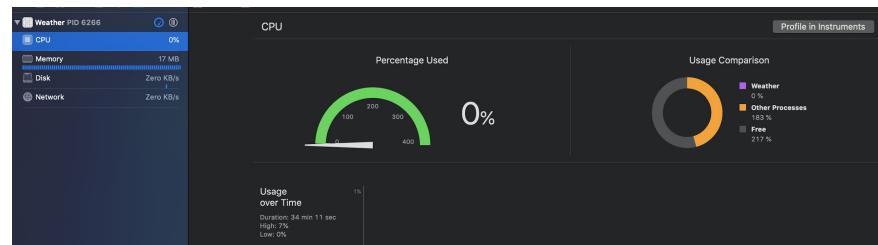


L'application demande beaucoup de ressources lors d'un clic



L'application

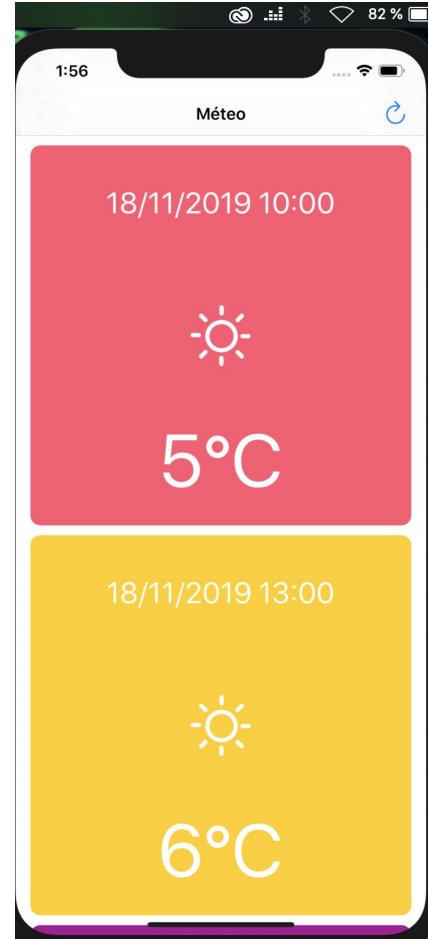
Au bout de 30min d'utilisation, nous constatons une utilisation une utilisation de 6,6Go de mémoire utilisée



Interface graphique

Le premier lancement de l'application est assez long puisque toutes les informations sont chargées à ce moment là

Lorsque nous coupons internet, les informations restent et notifie pas que les informations ne sont pas à jour. Il est nécessaire de rafraîchir l'application pour que les informations soient à jour.



Interface graphique

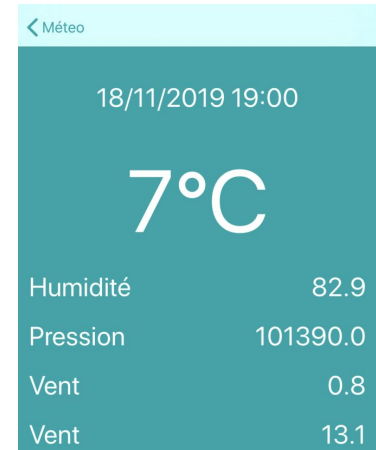
Pour plus d'informations, il suffit de cliquer sur le moment souhaité et une redirection affichera les détails

Chaque groupe de 4h possède une couleur différente

On constate également une absence d'options ou menu

Pas optimisé, un affichage d'une liste de 10 données est préférable (ou par jour).

Il télécharge les données une première fois, puis les stocks dans la RAM

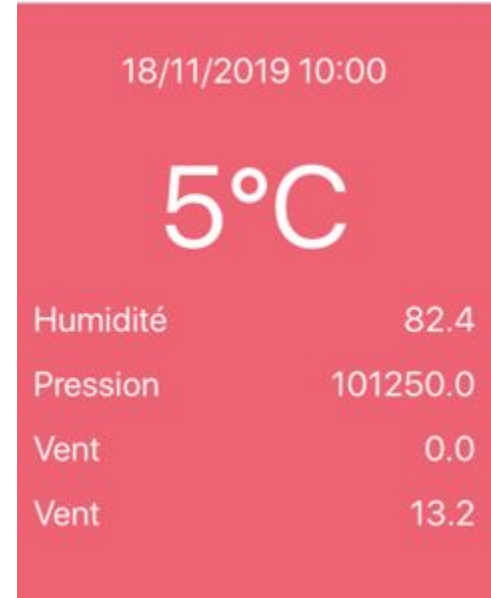


Interface graphique

Nous constatons que l'application ne montre pas d'unité

L'information sur le vent est mentionnée à deux reprises avec des valeurs différentes

L'information de la pression nous est inutile puisqu'elle n'est pas utilisée par l'utilisateur lambda



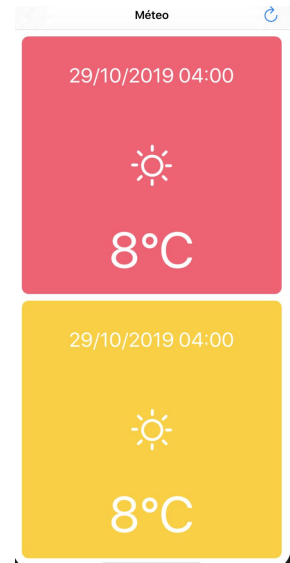
Interface graphique

Nous constatons également que l'icône du soleil est utilisée dans tous les cas

Une redondance des données est présente. Nous avons des informations pour une même heure pour une même date qui sont répétées

Nous ne savons de quel lieu viennent les données

Les dates sont décroissantes



Structure du code

Très peu de commentaires permettent d'expliquer les méthodes

La localisation est en brut, il n'y a donc pas de récupération de la météo selon la position courante

Pas de NetworkProvider les appel api sont directement initiés dans le modèle. Il est préférable de passer par une classe

Structure du code

Le projet est bien organisé, mais possède beaucoup de fonctions inutiles et vides.

Nous pouvons également retrouver des bibliothèques présentes et lourdes, mais pas utilisées.

```
23 func applicationWillResignActive(_ application: UIApplication) {
24     // Sent when the application is about to move from active to inactive state. This can occur for certain types of
    // temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and
    // it begins the transition to the background state.
25     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should
    // use this method to pause the game.
26 }
27
28 func applicationDidEnterBackground(_ application: UIApplication) {
29     // Use this method to release shared resources, save user data, invalidate timers, and store enough application state
    // information to restore your application to its current state in case it is terminated later.
30     // If your application supports background execution, this method is called instead of applicationWillTerminate: when
    // the user quits.
31 }
32
33 func applicationWillEnterForeground(_ application: UIApplication) {
34     // Called as part of the transition from the background to the active state; here you can undo many of the changes
    // made on entering the background.
35 }
36
37 func applicationDidBecomeActive(_ application: UIApplication) {
38     // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was
    // previously in the background, optionally refresh the user interface.
39 }
40
41 func applicationWillTerminate(_ application: UIApplication) {
```

Conclusion