

Rapport CLSW

POULAIN – MARACHE

Introduction et détails techniques

Public visé :

Sportifs occasionnels (tels que coureurs ou skieurs) souhaitant obtenir des statistiques sur leur pratique sportive.

Description du projet :

Le projet consiste en une application WearOS (SmartWatch), qui collecte des données du GPS ainsi que le rythme cardiaque, pour les traiter, les envoyer à un back-end et en afficher un résumé sur la montre.

Une autre application, Web cette fois-ci, permet de visualiser un dashboard qui affiche les données du back-end sous forme de graphiques et de cartes.

Choix technologiques :

- WearOS : Application Java, développement sous Android Studio
- Web : Vue JS (Vue 3), Vite, ApexChart (graphiques), Google Maps (carte)
- Back-End : Firebase Realtime Database

Protocoles de communication :

- WearOS ⇔ Back-End : API REST (Côté client : Retrofit)
- VueJS ⇔ Back-End : API REST (Côté client : Axios)

Application WearOS

L'application WearOS récupère 2 types de données :

- Les données des capteurs, dans notre cas le rythme cardiaque sous forme d'un float ;
- Les données de géolocalisation sous forme d'un objet Position.

Pour le rythme cardiaque, une méthode écoute les changements de valeurs du capteur et l'actualise en cas de déclenchement.

En ce qui concerne la géolocalisation, l'application écoute l'objet `FusedLocationProvider` qui déclenche un callback toutes les 3000ms. Ce callback effectue plusieurs actions :

- Calculer la distance entre la nouvelle position et l'ancienne, et l'ajouter à la distance totale du trajet en cours ;
- Extraire les données à envoyer au backend de l'objet `Position`, afin de sélectionner uniquement les valeurs qui nous intéressent : latitude, longitude, altitude, vitesse ; et y ajouter le dernier rythme cardiaque relevé.
- Envoyer les données au back-end.

Pour des raisons pratiques lors du développement, l'arrêt de la récupération de la position GPS lorsque la montre entre en `Ambient Mode` a été désactivée. Or, pour une version de production, il nous semble plus judicieux de l'activer.

Application Web

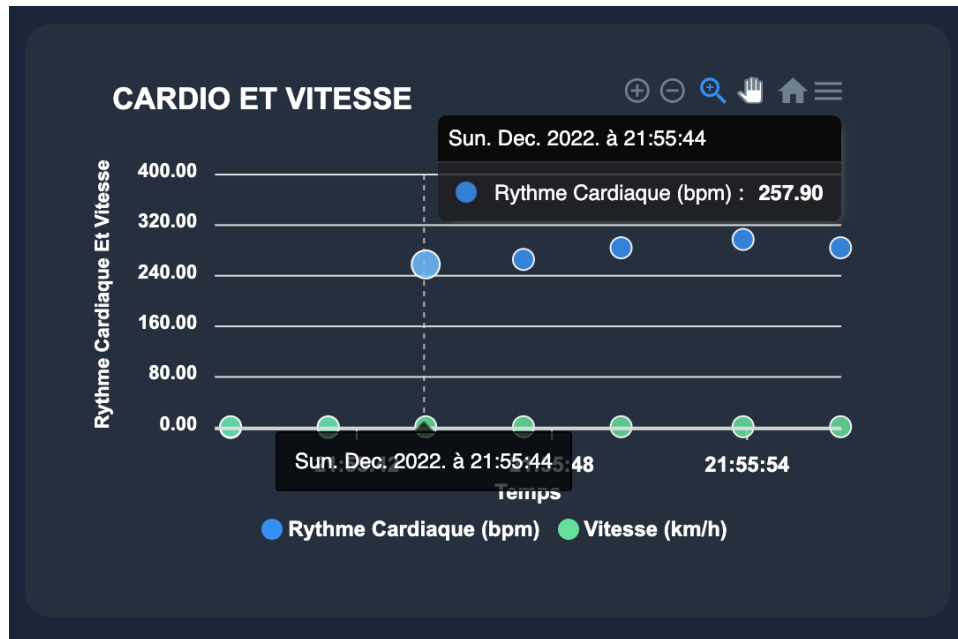
L'application Web permet de visualiser l'historique des vitesses, rythmes cardiaques et altitudes de l'utilisateur. Un premier graphique permet d'avoir une vue d'ensemble de cet historique en ce qui concerne la vitesse et rythme cardiaque (ces deux paramètres étant liés, pour un coureur par exemple), tandis qu'un second affiche les altitudes.



(Les points qui paraissent correspondre à la même date ne le sont pas, il faut zoomer sur le graphe.)

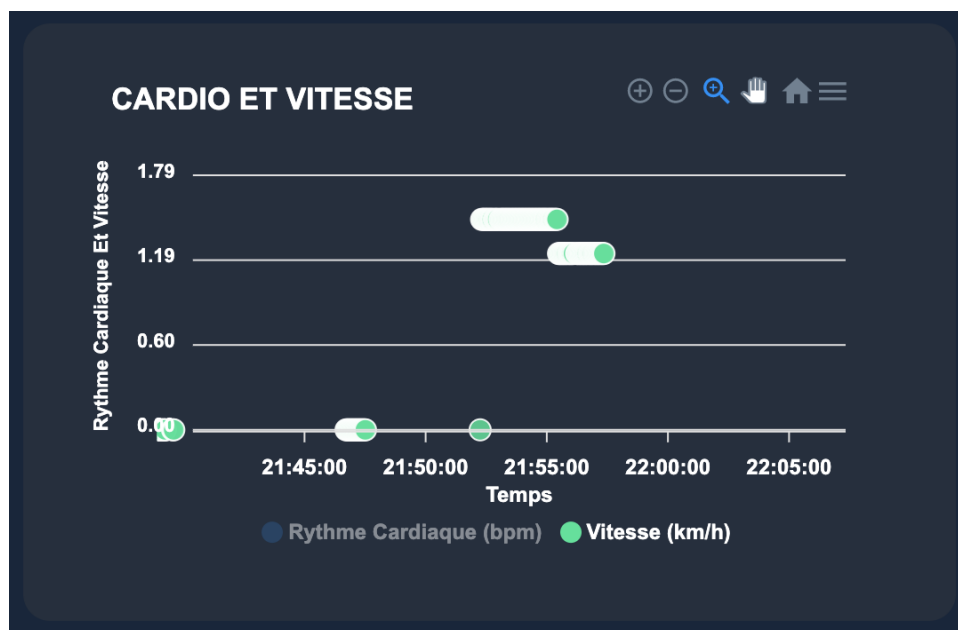
Ces deux graphiques sont interactifs, l'utilisateur peut :

- Zoomer/Dézoomer (en cliquant sur les boutons + ou -)
- Hover ou Sélectionner un point en particulier pour faire apparaître une tooltip affichant la date précise et les valeurs correspondantes

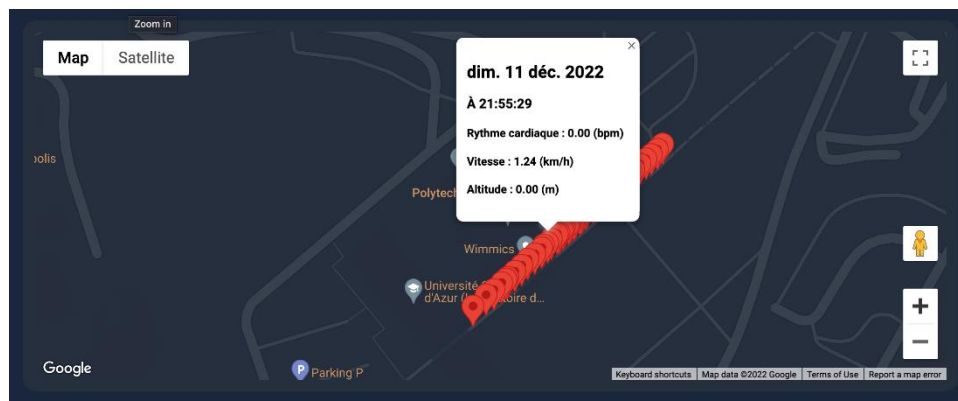


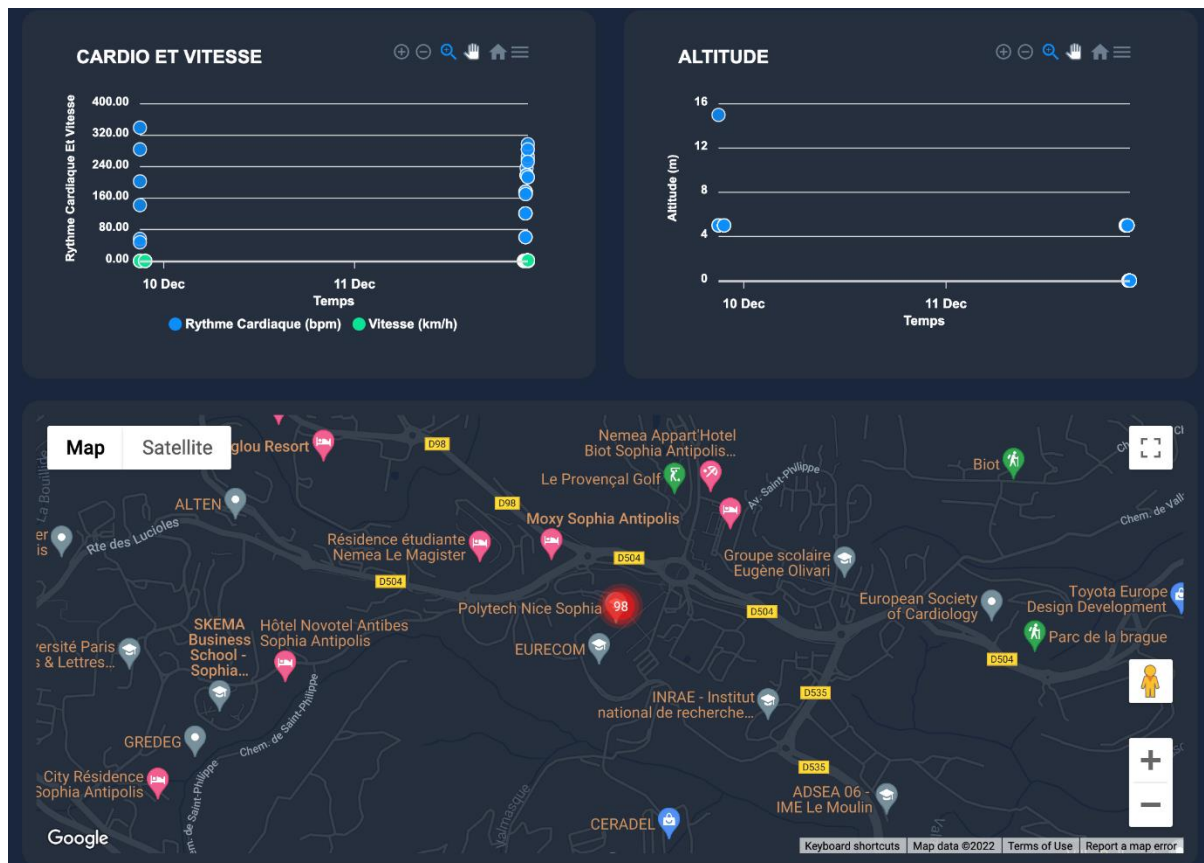
- Se déplacer dans le graphe (sur l'axe des abscisses) par drag (en cliquant au préalable sur l'icône "main")
- Sélectionner précisément une portion de graphe (clique et drag) pour redéfinir les extrémités de l'axe des abscisses (en cliquant au préalable sur l'icône de loupe)
- Revenir à la vue initiale (par défaut) en cliquant sur l'icône "home"

- Hover/Sélectionner/Désélectionner une série pour n'afficher qu'un des deux paramètres (vitesse ou rythme cardiaque)



Par ailleurs, une carte permet de visualiser les positions enregistrées par la montre, avec un détail sur le rythme cardiaque, la vitesse et l'altitude pour chacune d'entre elles. Pour afficher ce détail, il suffit de cliquer sur le repère (épingle) correspondant. À noter qu'une clusterisation des positions a été implémentée, pour voir plus distinctement les zones d'intérêt.





Points d'amélioration :

Si nous devions continuer le développement de ce mini-projet, il serait intéressant :

- De permettre à l'utilisateur de saisir précisément les dates qu'il veut consulter sur les graphes et de n'afficher que les positions correspondantes sur la carte
- Implémenter un algorithme permettant de détecter les trajets, dans le but de les "dessiner"
- Séparer les clusters de données en session, ce qui permettrait de mieux visualiser les données, et surtout obtenir un récap des sessions sur la montre.