Les langages cross-platform, également appelés langages de développement multiplateforme, sont des langages de programmation qui permettent de créer des applications qui peuvent être utilisées sur différentes plateformes et systèmes d'exploitation, tels qu'Android et iOS.

Voici quelques-uns des langages cross-platform les plus courants :

* **Flutter** : Flutter est un framework open-source développé par Google qui permet de créer des applications mobiles pour Android et iOS, ainsi que pour le web. Il utilise le langage de programmation Dart et offre une expérience de développement rapide et efficace.
* **React Native** : React Native est un framework open-source développé par Facebook qui permet de créer des applications mobiles pour Android et iOS en utilisant le langage de programmation JavaScript. Il utilise également la bibliothèque React pour la création d'interfaces utilisateur.
* **Xamarin** : Xamarin est une plateforme de développement multiplateforme appartenant à Microsoft qui permet de créer des applications pour Android, iOS et Windows en utilisant le langage de programmation C#. Il offre également des outils de développement avancés tels que des émulateurs et des outils de test.
* **Apache Cordova**: Apache Cordova est également un framework de développement d’applications multiplateformes open-source. Il est basé sur des technologies web telles que HTML, CSS et JavaScript, et permet de créer des applications mobiles pour plusieurs plateformes, notamment Android, iOS et Windows Phone.

Les avantages des langages cross-platform sont nombreux :

1. **Économie de temps et d'argent** : Avec les langages cross-platform, les développeurs peuvent créer des applications pour plusieurs plates-formes en utilisant un seul codebase, ce qui réduit considérablement le temps et les coûts de développement.
2. **Développement plus rapide** : Les outils de développement proposés par les langages cross-platform permettent une création rapide d'interfaces utilisateur et de fonctionnalités.
3. **Mise à jour plus facile** : Comme il n'y a qu'un seul codebase pour plusieurs plateformes, les mises à jour peuvent être effectuées simultanément pour toutes les plateformes, ce qui permet de gagner du temps et d'éviter les erreurs.
4. **Facilité de maintenance** : Avec une seule base de code, les mises à jour et les corrections de bugs peuvent être facilement appliquées à toutes les plateformes en même temps, ce qui simplifie la maintenance de l'application.
5. **Plus grande portée** : En développant une application cross-platform, on peut toucher un public plus large en la rendant disponible sur plusieurs plateformes.

Tout d'abord, j'ai opté pour le framework Cordova, car il permet de coder une application mobile comme une application web standard. J'ai développé une première version de l'application pour montrer à M. Burniaux l'avancement et pour vérifier si cela répondait à ses attentes. Cette première ébauche ne comportait qu'une interface utilisateur de base, sans aucune logique fonctionnelle, et pouvait être consultée depuis un navigateur web sur un ordinateur.

Développant sous Microsoft Windows j’ai pu remarquer que lorsque je souhaitais compiler le code grâce au framework Cordova, il me manquait un certains nombre de dépendance propre à Apple. Mr Burniaux me conseilla alors de me tourner vers un autre langage cross-platform, comme React Native qui a un module qui s’installe de base sur les appareils mobiles afin d’émuler un environnement stable.