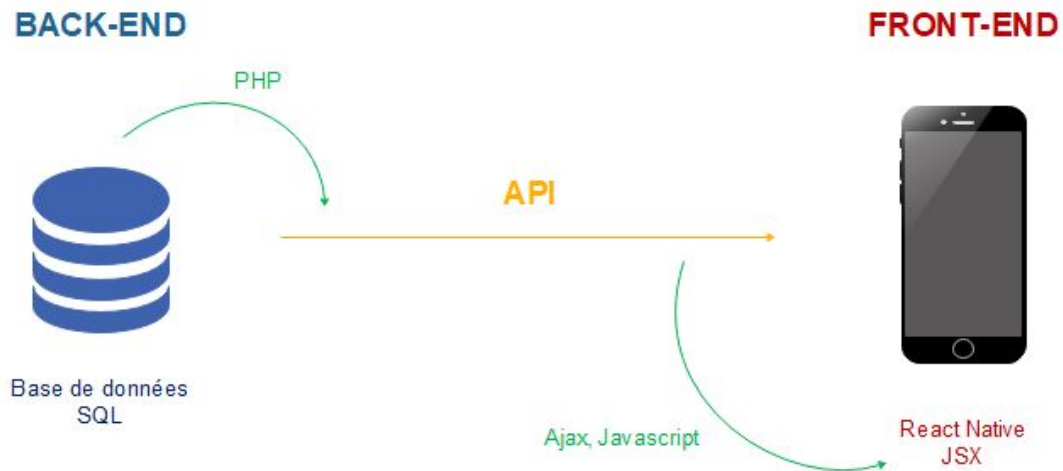


ARCHITECTURE LOGICIELLE



Nous avons choisi de réaliser notre projet sous la forme d'une **application mobile** car celle-ci possède de nombreux avantages comparé à une application web. Une application mobile est en effet plus pratique puisqu'il n'y a pas besoin d'un ordinateur pour s'en servir. Étant donné que notre service est un outil de gestion d'évènements, les utilisateurs seront certainement souvent en déplacement pendant qu'ils se serviront de l'application. Par exemple, au moment de faire leurs courses utiles à l'évènement, ils se serviront de leur liste de course créée sur l'application. Pour de simples raisons pratiques, il est préférable qu'ils puissent utiliser leur téléphone portable plutôt que leur ordinateur à ce moment-là.

Réaliser une application mobile implique de créer une API grâce à laquelle nous pourrions récupérer les informations stockées en base de données. Nous avons ainsi séparé notre architecture logicielle en **3 parties** :

1. La partie **BACK-END** :

Les données de l'application seront stockées dans une base de données relationnelle SQL. Celle-ci permet en effet de distinguer chaque entité et de bien modéliser les relations entre chacune d'entre elle.

2. **L'API**

Les données seront affichées en PHP car il est possible d'utiliser la notion de programmation objet tout en interagissant avec la base de données. Cela nous fera gagner du temps de développement : la plupart des composants de l'application seront modélisés sous forme de classes. En utilisant le PHP, nous pourrions encoder les données au format JSON pour pouvoir ensuite les afficher.

Nous avons choisi d'allier des requêtes Ajax à Javascript pour récupérer les données côté front.

3. La partie FRONT-END

Nous avons décidé d'utiliser React Native pour la partie front-end pour son adaptabilité, que ce soit pour des mobiles ou bien des tablettes. En effet, elle reprend les composants natifs à la fois d'iOS et d'Android.