Question Théorique

1. Car le code React native créer peut-être compiler en direction d’une plateforme en particulière. Ex : on peut compiler le code pour aller sur la plateforme IOS, Android et le web. Lors de la compilation on va pouvoir spécifier la cible.
2. Un développeur mobile doit se préoccuper de la compatibilité inter plateforme de son application. Car même si React native est capable de compiler le code pour plusieurs plateformes, il y a certaines choses qui ne vont pas fonctionner donc il faudra écrire du code dit natif afin de le permettre. La mémoire est aussi une chose à prendre en compte, les ordinateurs étant beaucoup plus performant, il va donc de soi de créer un code mobile qui soit le plus optimiser possible afin de garder la réactivité de l’application.
3. Il faudrait comme dit précédemment créer un code dit natif afin de faire correspondre l’action possible à l’os sur lequel seras exécuter l’action.
4. Les tests d’instantanés sont un outil très utile lorsque l’on souhaite s’assurer que notre interface utilisateur ne change pas de manière inattendue. Un cas de test snapshot classique affiche un composant d’interface utilisateur, prend un instantané, puis le compare à un fichier d’instantané de référence stocké avec le test. Le test échoue si les deux instantanés ne correspondent pas : soit la modification est inattendue, soit l’instantané de référence doit être mis à jour vers la nouvelle version du composant d’interface utilisateur.
5. useEffect permet de déclencher une action pendant un des cycle de vie du composant et prend fin quand le composant est fini de charger. Globalement ce hook permet ainsi de faire différentes actions au démarrage du composant.
6. Oui on peut faire en sorte que ce hook soit déclencher sur une action ou une valeur, que l’on appelle alors dépendance. Cela fera que le hook s’exécuteras si une valeur est changée ou si une action particulière est déclenchée.
7. On peut configurer des règles d’accès pour l’application utilisant Firebase en créant des règles d’accès directement sur Firebase, comme des règles d’accès à la base de données en temps réel, ou sur le stockage par exemple.
8. Il est avantageux d’utiliser Firebase car c’est un produit que l’on appelle un « backend as a service », ce qui veut dire que google à créer ce service afin de rendre disponible un backend sécurisé sans avoir à faire que très peu de chose afin que cela fonctionne avec notre application. Donc c’est avantageux car l’on gagne du temps. En utilisant Firebase que de créer un backend nous-même.
9. Faire du code natif permet de rendre disponible des fonctionnalités que React native ne fournit pas pour les deux OS IOS et Android.
10. Je ne sais pas vraiment à vrai dire, cependant je dirais que l’avantage d’expo c’est qu’il propose une suite d’outils que React-Cli ne propose pas, cependant React-Cli offre une plus grande latitude d’action alors qu’Expo est un peu bloquant des fois comme par exemple sur les build.