

Project dev mobile React Native : Daily Diet

Mabsout Oussama - Ruben Bouledjoudja

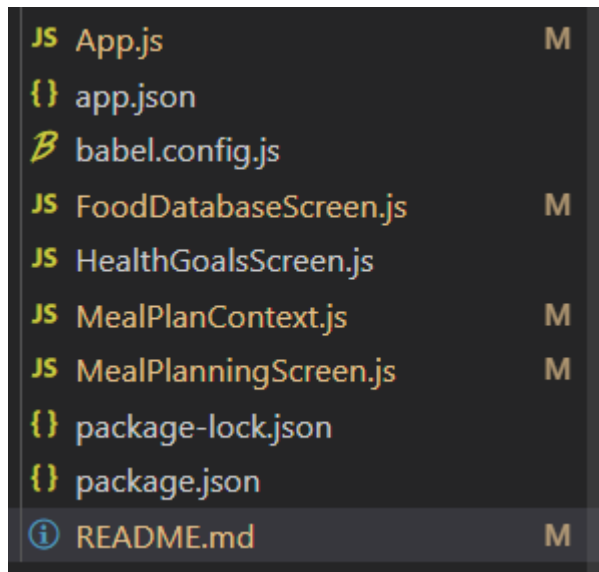
Objectives :

Etant donné que c'est notre première app mobile on a essayé de se tenir au consignes et de ne pas trop s'écarter de ces dernières afin de compléter le plus fidèlement le cahier des charges. L'objectif a été de créer un petit assistant de nutrition afin de cibler une perte de poids ou une prise de masse, l'outil doit rester le plus simple d'utilisation possible mais doit comporter tout le nécessaire.

Application's architecture:

L'architecture est assez simple :

Le fichier app.js comporte le module 1 qui a été assez simple car totalement vu en cours/td et nous a permis de créer le navigation tab mais il permet aussi d'initialiser le meal plan, après nous avons les 3 screens qui sont les 3 pages développées le health goal screen, la food database et le meal planning. et enfin le meal plan context est le gestionnaire du meal plan qui nous permet de le faire transiter d'une page à l'autre ainsi que de le garder en mémoire.



JS	App.js	M
{ }	app.json	
B	babel.config.js	
JS	FoodDatabaseScreen.js	M
JS	HealthGoalsScreen.js	
JS	MealPlanContext.js	M
JS	MealPlanningScreen.js	M
{ }	package-lock.json	
{ }	package.json	
i	README.md	M

Features:

Quasiment toutes les features du cahier des charges ont été implémentées avec succès. Sur toutes nos pages nous avons fait le choix d'intégrer le [TouchableWithoutFeedback](#) qui nous permet de quitter le clavier simplement mais sinon tout le reste est assez classique, l'ajout du scroll view...

Pour les modifications plus poussées nous avons tout d'abord l'utilisation de dropdown au lieu de picker dans le module 2(health goals) car nous avons trouvé que c'est beaucoup plus ergonomique et il nous posait moins de problème sur iphone.

Après une étude de l'api et du retour de la requête get que l'on utilise on a choisit d'afficher tout les element de liste retour car si l'on affiche que le premier élément reçu on obtient rarement ce que l'on recherche (si on recherche banana ou apple on tombe sur une marque ou des éléments modifié) donc pour avoir plus d'information sur l'aliment en question on appuie dessus afin d'avoir un aperçu (image plus calorie).

les éléments que l'on a trouvé important dans la database sont le nom l'image les calories et la quantité afin de se donner un ordre d'idée des calories contenues dans les aliments.

Le plus frustrant avec cette étape est que pas mal d'images dans la database sont manquantes ou assez mal prises donc cela nuit un peu à la qualité de l'affichage.

Les éléments que l'on a trouvé important dans la database sont le nom l'image les calories et la quantité afin de se donner un ordre d'idée des calories contenues dans les aliments. Et nous avons ajouté un petit calculateur qui permet de savoir la quantité avec l'unité de la database dans modal de la sélection des aliments.

La plus grosse épreuve a été de bien implémenter proprement le transfert de données entre la food database screen et le meal planning screen. on a réussi en créant le fichier mealplancontext qui va gérer le meal plan et qui va se charger aussi de la sauvegarde grâce au async storage.

Mais les principales features ont surtout été des features demandées qui sont réutilisées par exemple la désactivation du bouton add quand on a pas sélectionné de quantité dans la food database.

Pour les features que nous n'avons pas réussi à ajouter, il y'a le fait d'appuyer sur le bouton add d'un meal en particulier dans la page meal planning et de l'ajouter directement sans resélectionner le jour et le repas mais même si l'on arrive a envoyer en paramètre le jour et le meal il faudrait créer un autre modal qui aurait un affichage différent et donc choisir quel modal afficher. Cette feature est la seule feature obligatoire que nous n'avons pas réussi à réaliser.

Pour le style nous avons décidé de rester sur quelque chose de très classique, en utilisant quelques icônes de la librairie [Ionicons](#) et en gardant les mêmes tons de couleurs sur toutes les pages pour avoir un rendu propre et assez présentable.

Nous avons pensé à d'autres features a ajouter s' il nous restait plus de temps comme par exemple l'affichage en couleur du total calorique journalier dans le meal planning qui change si il dépasse un certain seuil et qui sert de baromètre afin de rendre l'application encore plus ergonomique. Mais la principale feature manquante est quand même l'écran d'accueil qui rendrait l'app bien meilleure

Conclusion :

Globalement ce projet fut une expérience très enrichissante comme introduction au développement mobile, il n'y a pas eu de grosse difficultés car le projet était très bien guidé les cours et td donnent le principal des informations nécessaire a sa realisation et en cas de manque d'information il est très facile de trouver des documentations très complètes sur internet.