

# **Compte Rendu Développement Mobile**

ESCAFFRE Robin - TADRES Nicolas

## Objectif

L'objectif de ce projet est de créer une application mobile permettant de calculer le nombre de calories et de planifier les repas pour un utilisateur.

## Architecture

L'application est composé de :

- Trois écrans contenu dans un dossier screens :
  - HealthGoals.js : affichant un formulaire pour calculer le nombre de calories par jour pour une personne en fonction de son âge, son genre, son poids, sa hauteur, son niveau d'activité, son objectif de santé
  - FoodDatabase.js : permettant d'effectuer une recherche sur les aliments et de planifier ces repas
  - MealPlanning.js : affichant le résumé des plats planifiés ainsi qu'une modification en ajoutant ou supprimant des plats
- Un fichier AppNavigator.js contenu dans un dossier navigation permettant de gérer la navigation entre les différents écrans.
- Un fichier FoodDatabaseService.js contenu dans un dossier services définissant la fonction permettant d'effectuer la recherche vers l'API Edamam.
- Un fichier storage.js contenu dans un dossier context contenant les fonctions permettant de gérer la façon dont nous stockons les données des utilisateurs.

## Fonctionnalités

Nous allons lister ici les fonctionnalités en décrivant les étapes et en expliquant le fonctionnement en fournissant des images.

- Formulaire pour le calcul du BMR (HealthGoals)
  - Affichage des champs (TextInput, Switch et Picker).

- Récupération des données saisie et utilisation des données pour le calcul du BMR (useState et création d'une fonction calculateBMR dans HealthGoals)
- Ajout d'un bouton qui est grisé si les champs ne sont pas remplis permettant de calculer le BMR

Sous IOS :

The screenshot shows the 'Health Goals' form on an iPhone. The form has the following fields: 'Age \*' (empty), 'Are you a male ? \*' (toggle switch off), 'Height \*' (empty), 'Weight \*' (empty), 'Activity Level \*' (dropdown menu showing 'Sedentary'), and 'Health Goal \*' (dropdown menu showing 'Weight Loss'). Below these fields is a greyed-out 'Calculate' button. At the bottom, the text 'BMR : 0 calories/day' is displayed. The bottom navigation bar includes icons for 'Health Goals', 'Food Database', and 'Meal Planning'.

*Capture d'écran avec formulaire vide*

The screenshot shows the 'Health Goals' form on an iPhone with the following filled fields: 'Age \*' (23), 'Are you a male ? \*' (toggle switch on), 'Height \*' (175), 'Weight \*' (75), 'Activity Level \*' (dropdown menu showing 'Light Exercise'), and 'Health Goal \*' (dropdown menu showing 'Weight Maintenance'). Below these fields is a blue 'Calculate' button. At the bottom, the text 'BMR : 2478 calories/day' is displayed. The bottom navigation bar includes icons for 'Health Goals', 'Food Database', and 'Meal Planning'.

*Capture d'écran avec formulaire et calcul BMR*

Sous Android :

The screenshot shows the 'Health Goals' form on an Android phone. The form has the following fields: 'Age \*' (empty), 'Are you a male ? \*' (toggle switch off), 'Height \*' (empty), 'Weight \*' (empty), 'Activity Level \*' (dropdown menu showing 'Sedentary'), and 'Health Goal \*' (dropdown menu showing 'Weight Loss'). Below these fields is a greyed-out 'Calculate' button. At the bottom, the text 'BMR : 0 calories/day' is displayed. The bottom navigation bar includes icons for 'Health Goals', 'Food Database', and 'Meal Planning'.

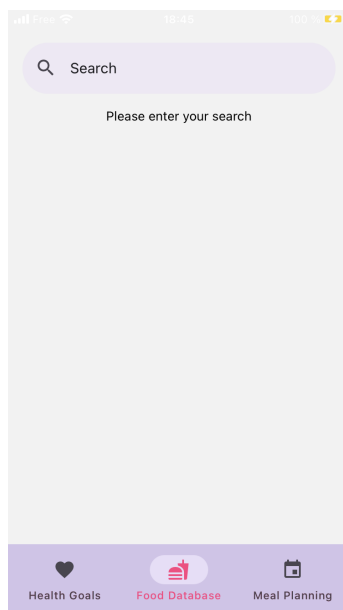
*formulaire vide*

The screenshot shows the 'Health Goals' form on an Android phone with the following filled fields: 'Age \*' (23), 'Are you a male ? \*' (toggle switch on), 'Height \*' (175), 'Weight \*' (75), 'Activity Level \*' (dropdown menu showing 'Light Exercise'), and 'Health Goal \*' (dropdown menu showing 'Weight Maintenance'). Below these fields is a blue 'Calculate' button. At the bottom, the text 'BMR : 2478 calories/day' is displayed. The bottom navigation bar includes icons for 'Health Goals', 'Food Database', and 'Meal Planning'.

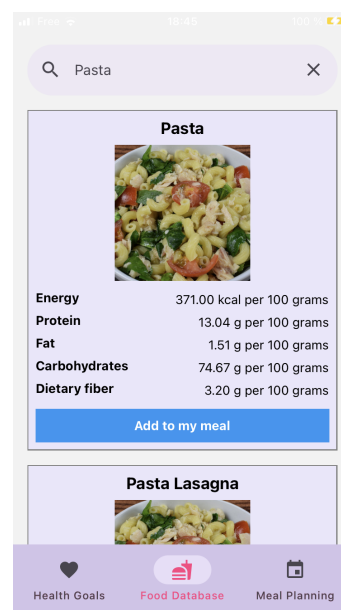
*formulaire plein*

- Recherche et ajout d'un aliment (FoodDatabase)
  - Affichage d'un champs permettant de taper la recherche en utilisant SearchBar + création de la fonction de recherche
  - Affichage de ce qui est retourné par l'API
  - Création d'un bouton "Add to my meal" ouvrant une modal qui contient des champs à remplir pour pouvoir ajouter l'aliment. Le champ "food" est un champ disabled, il permet juste de savoir quel plat nous avons sélectionné. Le champ "date" permet de sélectionner une date supérieure ou égale à la date du jour car nous ne voulons pas que l'utilisateur ajoute un aliment pour une date passé.
  - Création d'un bouton "Add" qui est grisé si les champs ne sont pas remplis et qui permet d'ajouter le plat (utilisation de AsyncStorage avec comme clé la date saisie par l'utilisateur)

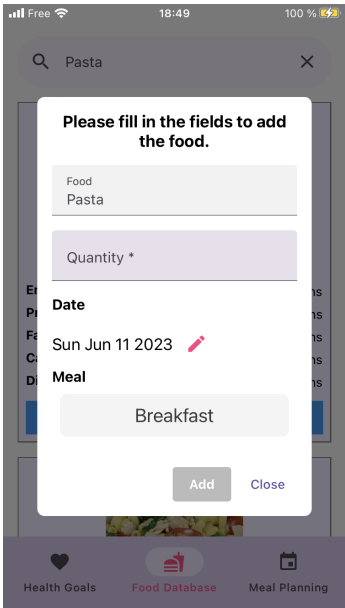
Sous IOS :



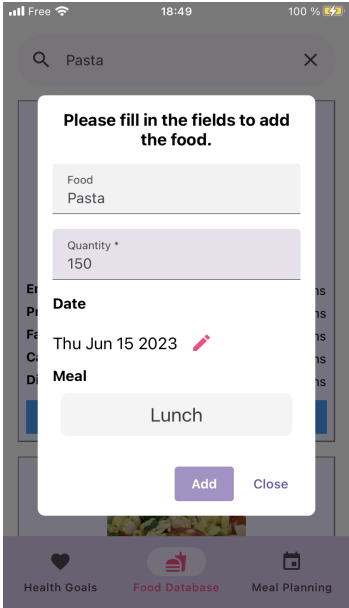
*Capture de l'écran de recherche sans saisie*



*Capture de l'écran de recherche avec saisie*

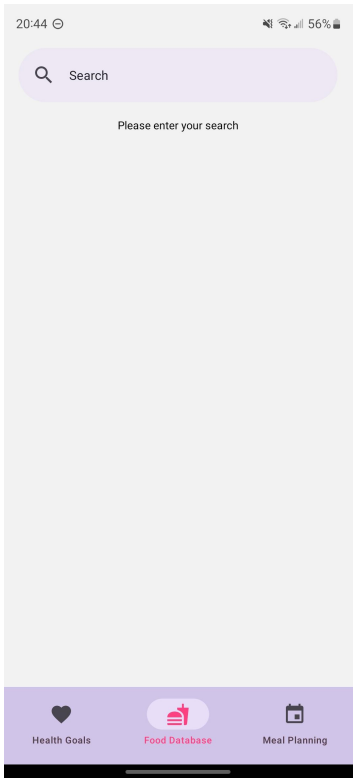


Capture de la modal non remplie

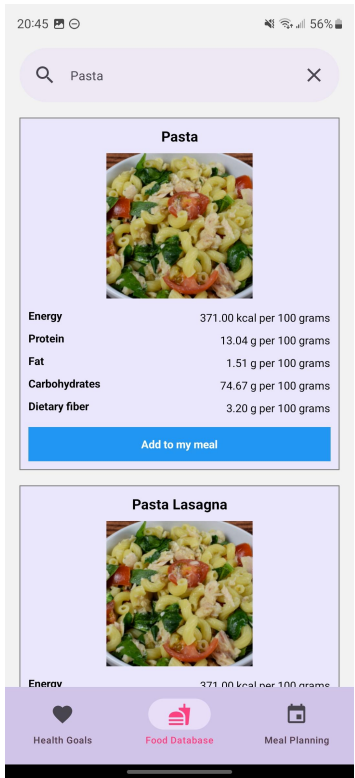


Capture de la modal remplie

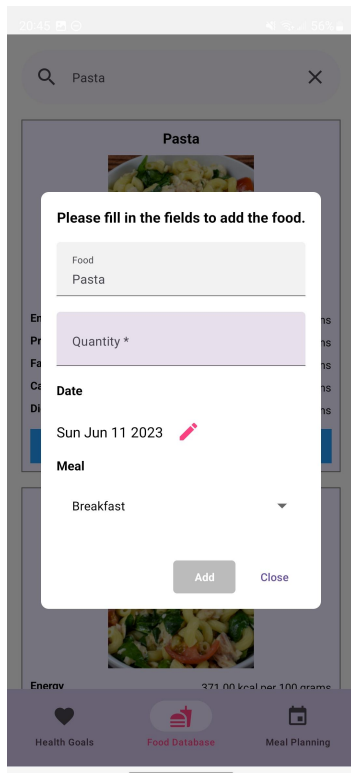
Sous Android :



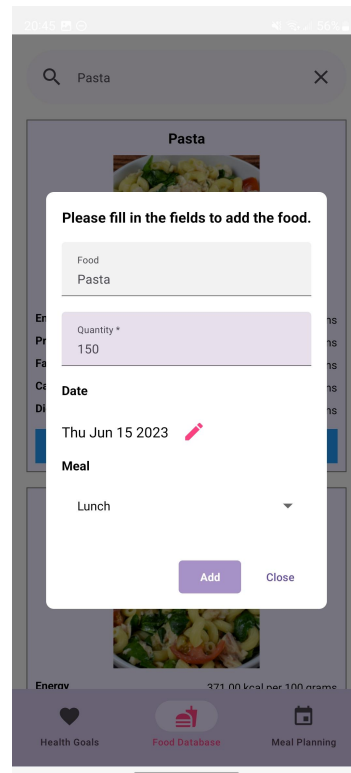
sans saisie



avec saisie



*modal non remplie*



*modal remplie*

- Planification des repas, calcul des calories et modification des repas (MealPlanning)
  - Affichage des résultats (récupération grâce à AsyncStorage et grâce à la date sélectionnée par l'utilisateur).
  - Ajout d'un bouton "delete" permettant de supprimer un aliment
  - Ajout d'un bouton "Add Food" permettant d'ajouter un aliment à partir de l'écran Meal Planning (utilisation de useRoute et useNavigation pour ajouter des paramètres dans les routes).
  - Ajout de conditions qui modifie la modal dans l'écran FoodDatabase en fonction de si on navigue vers FoodDatabase depuis le bouton présent dans MealPlanning ou depuis un autre composant

Remarque : l'utilisateur peut visualiser une date inférieure à la date du jour dans l'écran MealPlanning mais ne pourra pas ajouter ou supprimer d'aliments.

Sous IOS :

Fri Jun 16 2023

Breakfast

Nothing

Add Food

Lunch

Nothing

Add Food

Snack

Nothing

Add Food

Health Goals

Food Database

Meal Planning

MealPlanning vide

Sun Jun 11 2023

Breakfast

Food	Energy (kcal per 100g)	Energy for your quantity	Delete
Fried Egg	196.00	196 (100g)	X
Milk	61.00	91.5 (150g)	X

Total Energy : 287.5 kcal

Add Food

Lunch

Food	Energy (kcal per 100g)	Energy for your quantity	Delete
Pizza	268.00	938 (350g)	X

Dinner

Food	Energy (kcal per 100g)	Energy for your quantity	Delete
Salads Tuna Salad	246.48	1109.16 (450g)	X

Total Energy : 1109.16 kcal

Add Food

Total Energy for the day: 2360.66kcal

Health Goals

Food Database

Meal Planning

MealPlanning remplie

Sat Jun 10 2023

Breakfast

Nothing

Lunch

Food	Energy (kcal per 100g)	Energy for your quantity
Pizza	268	536 (200g)

Total Energy : 536 kcal

Snack

Nothing

Health Goals

Food Database

Meal Planning

Sat Jun 10 2023

Lunch

Food	Energy (kcal per 100g)	Energy for your quantity
Pizza	268	536 (200g)

Total Energy : 536 kcal

Snack

Nothing

Dinner

Nothing

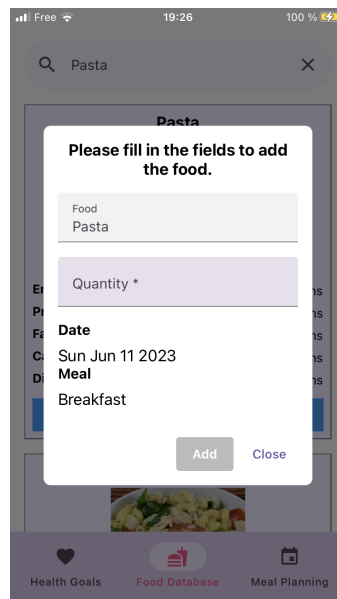
Total Energy for the day: 536kcal

Health Goals

Food Database

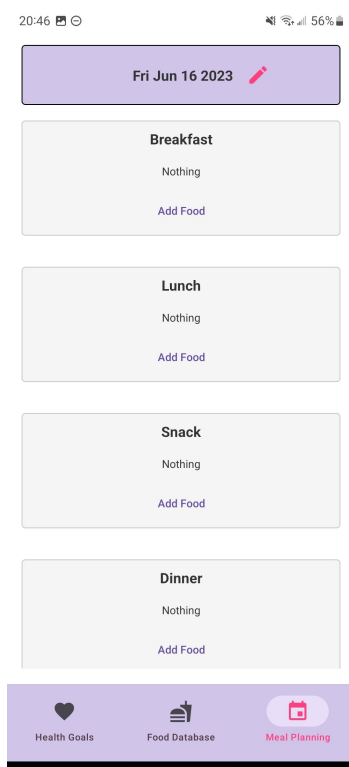
Meal Planning

MealPlanning d'une date passé (la date des captures d'écran était le 11/06)

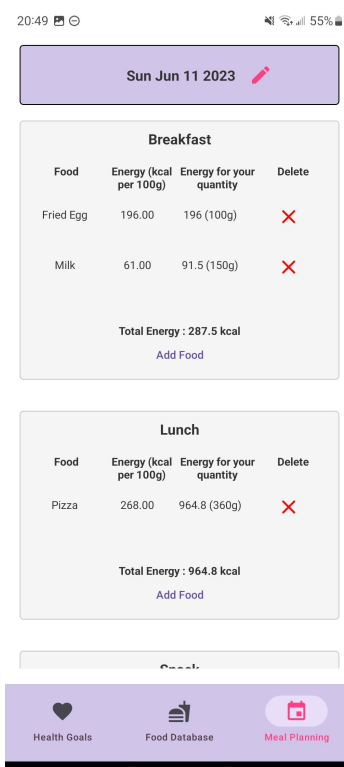


*Capture de la modal une fois qu'on a appuyé sur le bouton "Add Food" de MealPlanning et qu'on a sélectionné le plat à ajouter. On remarque que le seul champ à remplir est le champ "quantity" les autres champs sont juste des informations qu'on a récupéré.*

Sous Android :

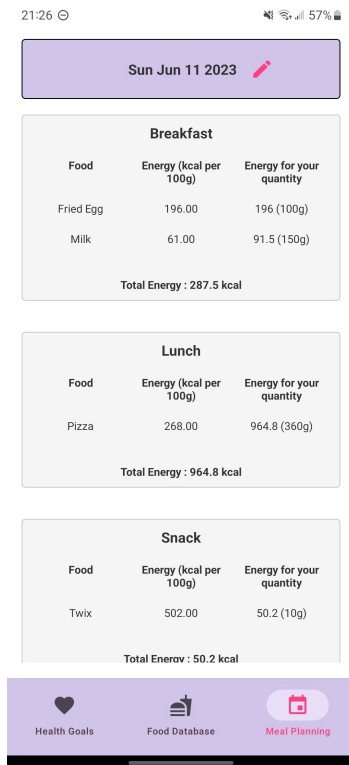


MealPlanning vide

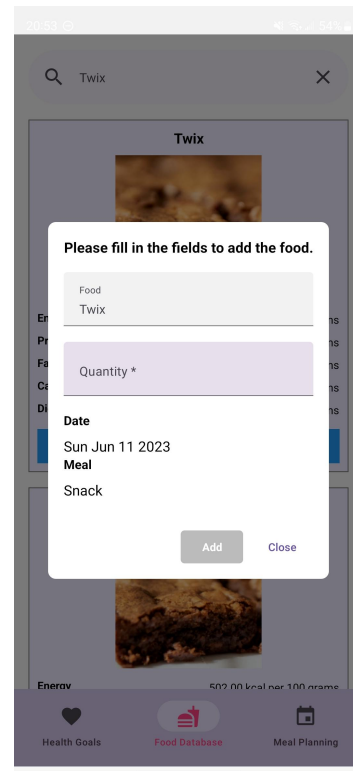


MealPlanning remplie





*MealPlaning date passée*



*Modal en passant par MealPlanning*

## Difficultés

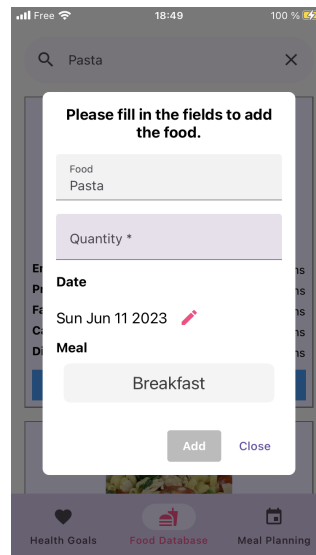
Nous avons rencontré 4 principales difficultés :

- Mise en place de l'API :  
Nous avons passé énormément de temps à comprendre comment l'API fonctionne car cela était tout nouveau pour nous. Nous avons également eu du mal avec les données retournées par cette API car certaines données contenait des images et d'autres non et fallait donc gérer les exceptions et les erreurs qu'on rencontrer en fonction des recherches effectuées dans le FoodDatabase.
- Le champ Date :  
Cela nous a posé énormément de problèmes. En effet, nous avions voulu suivre la correction du TD 4 en utilisant le DateTimePicker pour avoir un champ Date mais cela fonctionnait sur IOS et non sur Android. Nous en avons énormément parler avec Madame LAMBERT Tiana et après avoir consulter la documentation (<https://github.com/react-native-datetimepicker/datetimepicker>) nous avons vu qu'il fallait ajouter un paramètre qui rend visible ou pas le champs Date en ajoutant un bouton.

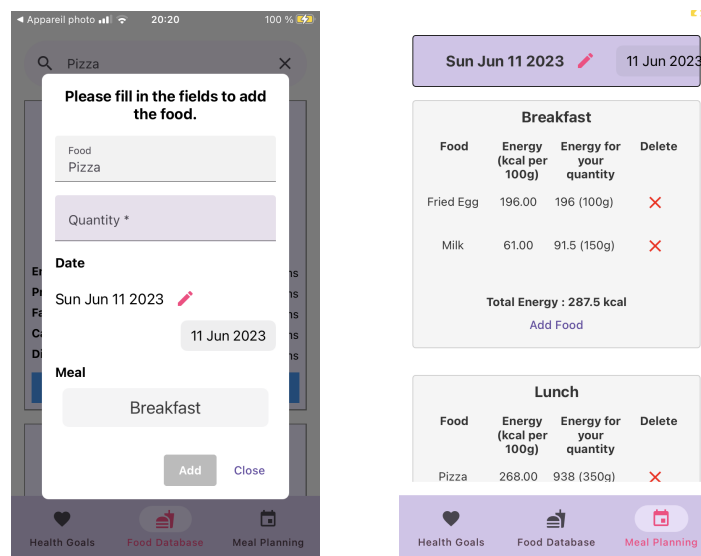
Cela était dû au fait que sous Android le champ DateTimePicker s'ouvre dans une modal.

La solution fonctionnait mais le rendu est "moins beau" sur IOS car, en appuyant sur l'icône pour modifier la date, on a un champ en plus qui apparaît ce qui n'est pas le cas sous Android où la modale s'ouvre directement. Comme nous pouvons le voir sur les images ci-dessous

Sous IOS :

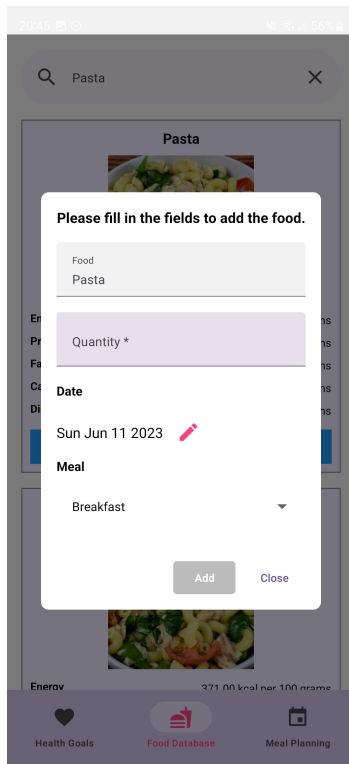


*Champ date sans avoir appuyer sur l'icône*

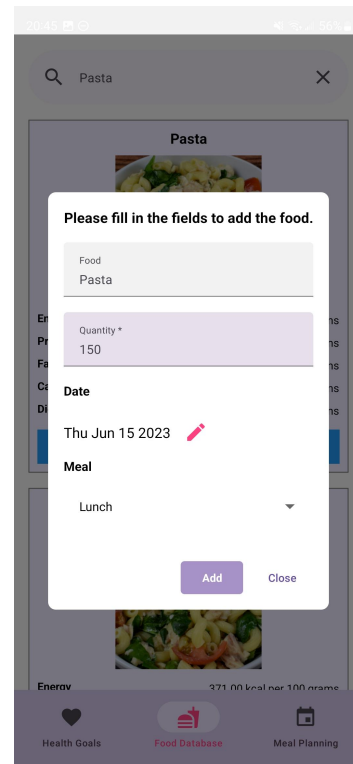


*Champ date après avoir appuyer sur l'icône*

Sous Android :



*date avant d'avoir changé*



*date après avoir changé  
(pas de problème)*

- Mise en place du AsyncStorage :  
Ici la difficulté était de comprendre comment cela fonctionnait et de trouver le meilleur moyen de sauvegarder les données (c'est-à-dire quelle clés et valeur choisir)
- Ajouter un aliment depuis le MealPlanning :  
Le plus dur était de savoir comment mettre à jour les paramètres passés dans la navigation une fois qu'on quitte l'écran FoodDatabase (en effet il fallait les rendre undefined). Pour cela nous avons ajouté un écouteur d'événement sur le "blur" qui va modifier les paramètres à undefined lorsque l'on quitte l'écran.

## Conclusion

Pour conclure nous aimerions parler des points positifs de notre application et les pistes d'améliorations que l'on peut apporter.

Points positifs :

- Cahier des charges respectés
- Application fonctionnelle sans bugs

Pistes d'améliorations :

- Style de l'application : nous avons essayé de stylisé l'application mais Robin et moi nous ne sommes pas très doué en CSS et en style / conception de façon générale
- Code : nous pensons que le code peut être un peu mieux organisé et pourquoi pas découper en plusieurs fonctionnalités.

Cependant, ce projet nous a permis de découvrir le développement mobile que nous n'avons jamais fait auparavant et nous avons compris plein de concepts intéressants.