##################################################################################

-Pour pouvoir tester l’application vous devez d’abord installer l’environement pour react native depuis ce lien : <https://reactnative.dev/docs/environment-setup>.

-Telecharger le projet dans votre local avec cette commande :

**git clone https://github.com/YuzuAdmin/Yuzu-beta-v2.git**

-Telecharger le package,selon votre manager package :

**Yarn** OR **npm install**

-Pour lancer android :

**Yarn android**

-Pour lancer IOS vous devez d’abord installer les PODS :

**Cd ios** && **pod install**

-Et puis pour le lancer :

**Yarn ios**

HAVE FUN !

##################################################################################

**Documentation**

Cette application est ecrite en React Native,elle est divisée en trois parties :

Front End + Back End + Application Web .

**-Back End :** lien Github du projet : <https://github.com/tokismos/backendYuzi> .

Les différentes technologies que J’ai utilisé pour créer la backend sont :

* Express Js : c’est une technologie qui nous permets de créer des API REST et c’est ce qui connectera notre back end avec le front end, c’est grâce à cette dernière qu’on pourrait lire et écrire les donnes dans la base de donne .

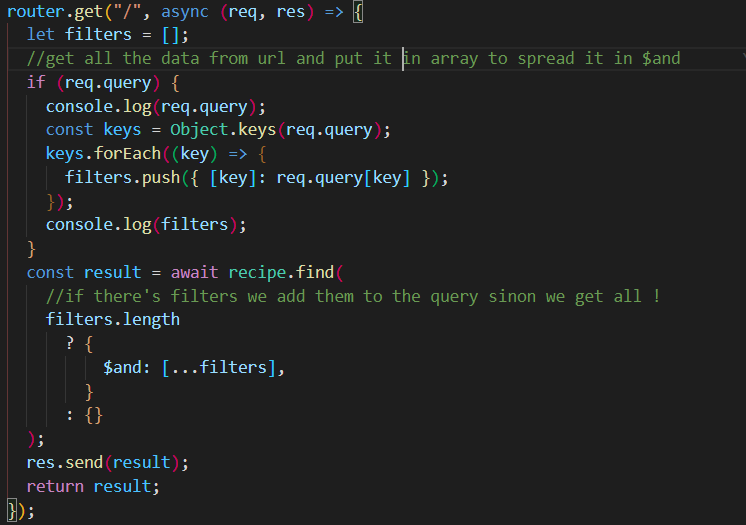
Jusqu’à présent elle se compose de deux routes :

* Un GET qui nous permets de récupérer les recettes de la BD , et cela en appelant ‘ /recipes ‘ ,par exemple pour notre cas : [https://backend-yuzi.herokuapp.com/recipes](https://backend-yuzi.herokuapp.com/recipes%20)  . Cette dernière nous renvoie toutes les recettes sous format JSON.
* Un POST qui nous permet d’ajouter dans la base de donnée des recettes, et cela en appelant ‘/recipes/add’ en envoyant les donnes dans le body ,par exemple pour notre cas :

[https://backend-yuzi.herokuapp.com/recipes/add](https://backend-yuzi.herokuapp.com/recipes/add%20)

* MongoDB : c’est la base de donne que j’ai utilisé ,j’ai choisi atlas MongoDB parce que ça facilite le déploiement et le management de cette dernière.

Voici la structure des recettes de notre BD : 

C’est ce bout de code qui nous permets d’afficher les recettes selon certains critères , par exemple si on veut ajouter des filtres :

Ce code veut dire que si on active un ou plusieurs filtres, on va spread tous ces derniers et on fait une requête de la bd avec ces conditions. Sinon s’il n’y a pas de filtres on affiche tout. Les filtres sont détectés automatiquement dans le lien de GET.

**-Application Web :**

-lien Github : <https://github.com/tokismos/BDWEB>

-lien du site : <https://yuziweb.herokuapp.com/>

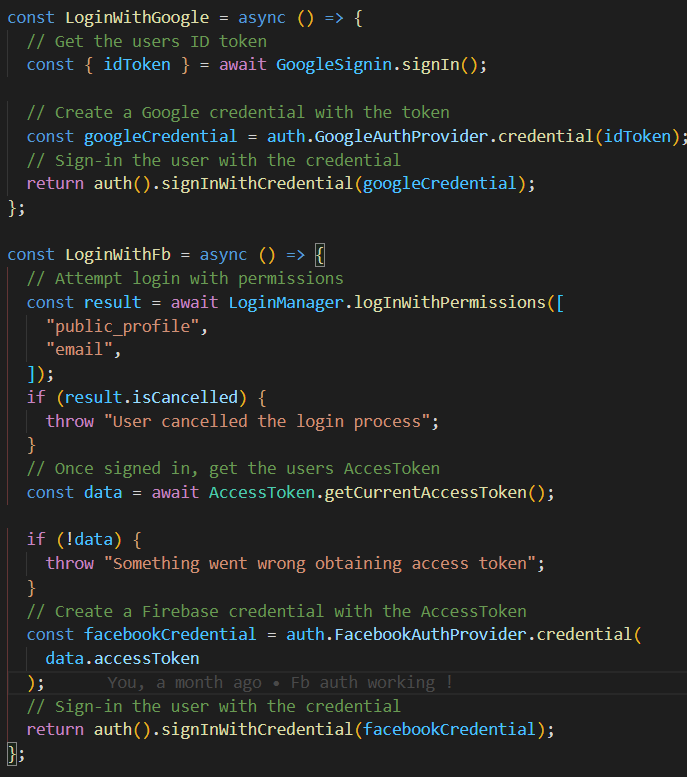
Ce site nous permets d’ajouter des recettes a la base de donne et cela d’une manière plus facile et rapide .Pour son fonctionnement je l’ai explique a tim.et Pour le code y’a pas quelque chose de spécial a expliquer.

**Front End :** Lien Github : <https://github.com/tokismos/foodApp>

Pour le premier écran qui s’affiche c’est le Onboarding screen ( src/screens/OnBoardingScreen.js) c’est un écran qui apparait la première fois que l’utilisateur installe l application, il permets d’expliquer cette dernière.

Apres c’est l’écran de login ,on peut se connecter soit avec email , facebook ,google,ou numero de telephone .

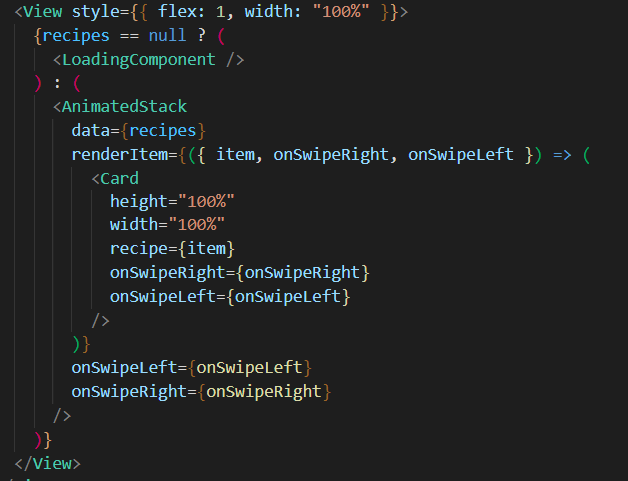
Voici le code de la connexion par Facebook et google : il se trouve dans src/helpers/db

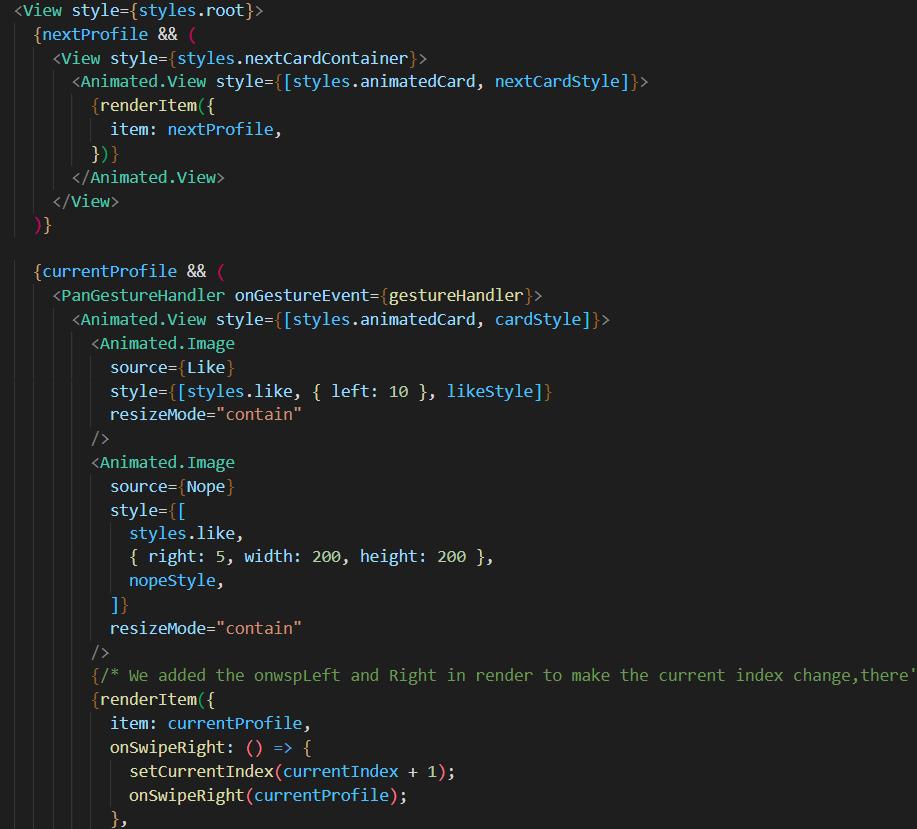


J’ai utilisé Firebase Authentification pour gérer cette partie, elle nous permettra de se connecter de manière sécurisée.

Apres cet écran on retrouve l’écran de SignUp, qui comme son nom l’indique nous permettra de s’inscrire, et c’est la ou on pourra introduire notre numéro de téléphone , dans lequel on va recevoir un code pour le confirmer.

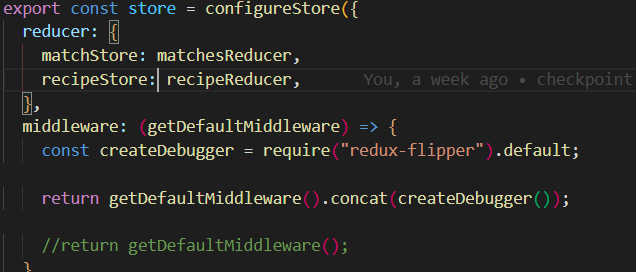
Apres s’être connecte la première page c’est TinderScreen c’est la ou on trouve les différentes recettes de la BD .



C’est ce code ci-dessus qui nous affiche ces cartes de Tinder ,il veut dire que si les recettes n’existent pas on affiche un loading sinon on affiche les cartes, et cela dans le component AnimatedStack , j’ai créé ce dernier comme la logique d’une FlatList.

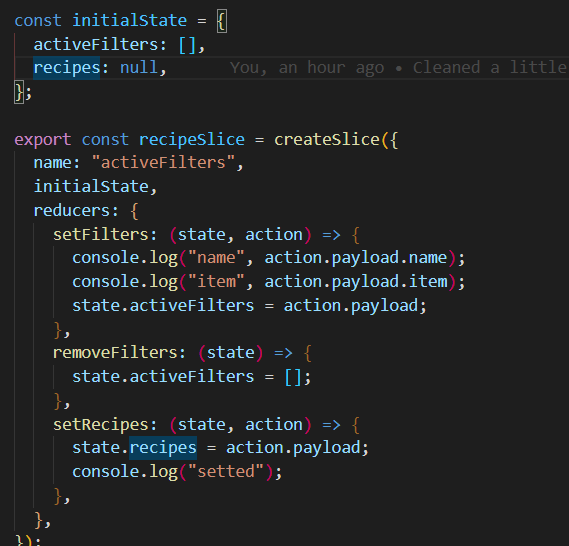
Il se compose en deux View la première et la deuxième qui affiche la recette qui suive.L’animation se fait en Reanimated 2.En swipant a gauche ou a droite on augmente l’index de chaque carte de 1 pour récupérer la valeur de la prochaine recette dans la liste.

Pour le filterScreen (src/screens/FilterScreen.js) c’est la qu’on choisissent les différents filtres a appliquer. Ces filtres doivent être sauvegardés dans une state de Lapp, c’est pour cela que j’ai utilise REDUX TOOLKIT ,il nous permet de gérer la state ainsi de passer cette dernière dans les differents components de l’app :

Il se compose de deux stores : 

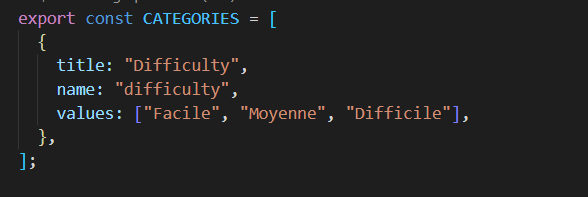
matchStore : C’est pour gerer les matchs.

recipeStore :c’est la ou on gere les differentes recettes ainsi les differents filtres :



Au debut les filtres actives sont vides, c’est avec setFilters qu’on purrait les ajouter dans l’Array,et setRecipes pour ajouter les recettes qu’on recoit de la BD

Pour les filtres on peut les ajouter dans le fichier ‘src/helpers/categories’ :

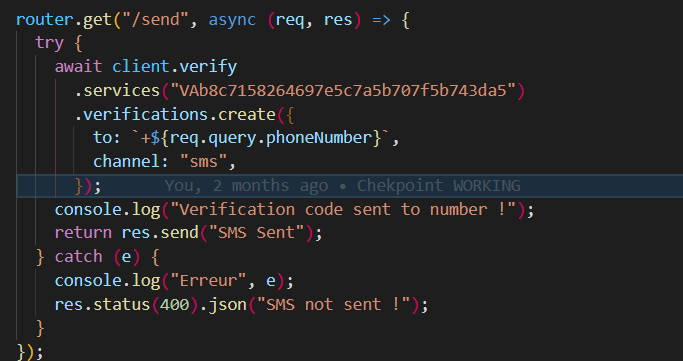


Apres avoir ajouter dans ce fichier, ces filtres sajoutent automatiquement dans lecran filterScreen,et on peut desormais les utiliser sans aucune configuration, qu’on on appele l’API ces derniers sont ajoutes automatiquement dans l’URL de GET , et la requete de la BD est faites a partir de ces filtres.

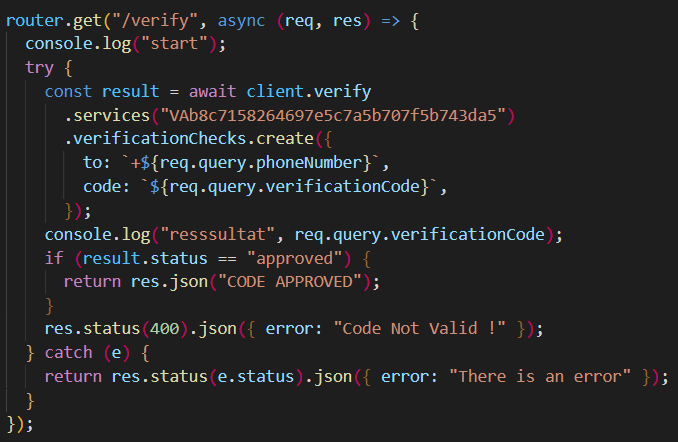
Dans cette section on trouvera une explication de chaque écran :

-Dans le dossier createAccountScreens on trouve **SignUpScreen** : C’est dans cet écran que se passe l’inscription, et il se constitue de quartes Views : l’email, le mot de passe, le numéro de téléphone ,confirmation de code. Pour la vérification du numero,Yuzu fait appel a l’api de Twilio qui nous permets d’envoyer un sms et vérifier sa validite,et cela se fait avec notre backend.

L’image ci-dessous est le code qui permet d’envoyer un message de vérification :



L’image ci-dessous est le code qui permet de vérifier la validité du code de vérification :

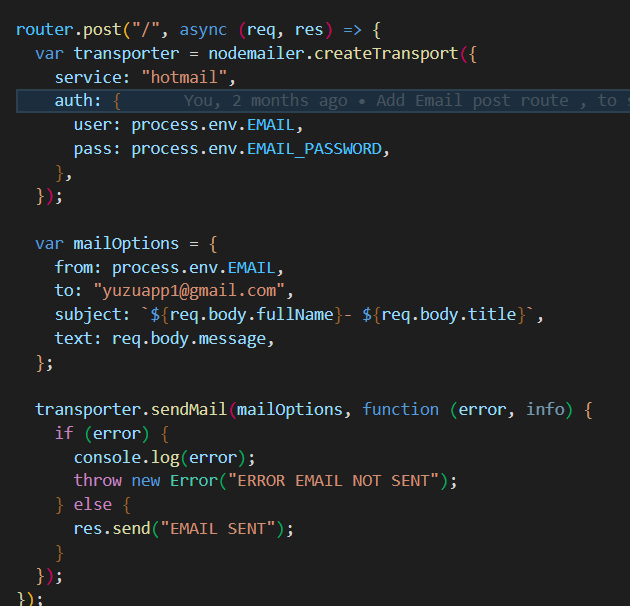


Le fichier **CommandeScreen** est l’écran de commandes passées par l’utilisateur, et ils disposent des ingredients aussi. Pour son fonctionnement c’est assez simple,y’a rien a expliquer.

Le ficher **InfoCommandeScreen** c’est la page qui affiche les ingrédients des recettes enregistrées.

Le fichier **FeedBackScreen** c’est la ou l’utilisateur pourrait envoyer un feedback vers l’adresse email designée,le message est envoyé via un api dans notre backEnd,ce package s’appelle nodeMailer et il nous permet d’envoyer un message a une adresse email.

Ceci est le code de cet api :



Le fichier **IngredientCartScreen** c’est la page ou on peut choisir les ingrédients de chaque recette enregistrées auparavant

Le fichier **IngredientScreen** c’est la page qui contiennent toutes les informations a propos de la recette,et elle s’ouvre lorsque l’on clique sur le bouton d’exclamations dans la première page TinderScreen.Cette page contient un param ‘params.\_id’,quand ce dernier existe elle permet d’afficher le bouton de vote en bas de la page :



Le fichier **IntroScreen** c’est la toute première page qui nous permet de se connecter,s’inscrire.c’est la page la plus simple a mettre en place dans ce projet.

Le fichier **MyRecipesScreen** c’est la ou on peut voir les différentes recettes enregistrées et on pourrait ainsi accéder a ses Details. On récupère les donnes de la recette en appelant notre Base de donné Firebase avec la fonction getCommandes() , cette dernière est enregistré dans notre document Firebase .

Le fichier **OnBoardingScreen** c’est le tout premier écran ou on peut voir une petite introduction ou tutoriel a propos notre application.

Le **PanierScreen** c’est la ou on a le choix de sélectionner les recettes qu’on a swipés,et cela en les cochant ou décochant.

Le fichier **PhoneScreen** c’est une page qui ressemble à la page précédente SignUpScreen,elle permet de changer le numéro de téléphone et cela en appelant l’api de TWILIO (Voir en haut ).

C’est dans l’écran **ProfileScreen** qu’on trouve les différentes informations a propos de l’utilisateur connecté, et il aura la possibilité de changer son nom ou numéro, et ainsi de future fonctionnalités…

Si l’utilisateur ne dispose pas d’un numéro de téléphone ,yuzu va devoir montrer une fenêtre a chaque fois pour demander a l’utilisateur de l’ajouter,parceque c’est une information très importante dans notre application.

L’écran **SignInScreen** nous permettra de se connecta via un email et mot de passe a partir de firebase , on trouvera cette fonction dans le fichier useAuth() dans le dossier hook.Et nous permettra aussi de se connecter via Google Auth.

L’écran **TinderScreen**  c’est l’écran principal de l’application c’est a partir de celui la qu’on pourrait swiper les recettes,naviguer vers les autres pages ,envoyer un feedback,ou accéder a son profil.Le code du swipe se trouve dans le dossier component dans TinderCard.js .