MiAM.

Projet mobile

HELB 2021-2022

Al haddaoui Marwane

RAPPORT D'ITÉRATION 3

INTRODUCTION :

Dans le cadre du cours de projet mobile, une application Android nous a été demandée d'être créé, vis-à-vis de la thématique de l'application mon choix c'est naturellement porté vers une application de Delivery Food (une application de livraison de nourriture), l'application se nomme miam ,elle a pour but de simuler une livraison de n'importe quel type de nourriture entre un client sur l'application et un restaurant ou un magasin. Elle proposera également une rubrique recette afin d'inciter l'utilisateur à l'achat des produits alimentaires autres que la nourriture fast fast-food(achat de légume, fruits, etc.)

FEEDBACK :

Au cours de la dernière rencontre, la structure globale des rapports à étaient approuvés ainsi que leur contenue, une suggestion d'un system de "PORT/Card" a été émise vis-à-vis des items à incorporer dans l'application, cependant le rapport devrait contenir également certains passages explicitant certaines difficultés rencontrées voire certaines modifications des outils, tels que l'émulateur modifié et personnalisé.

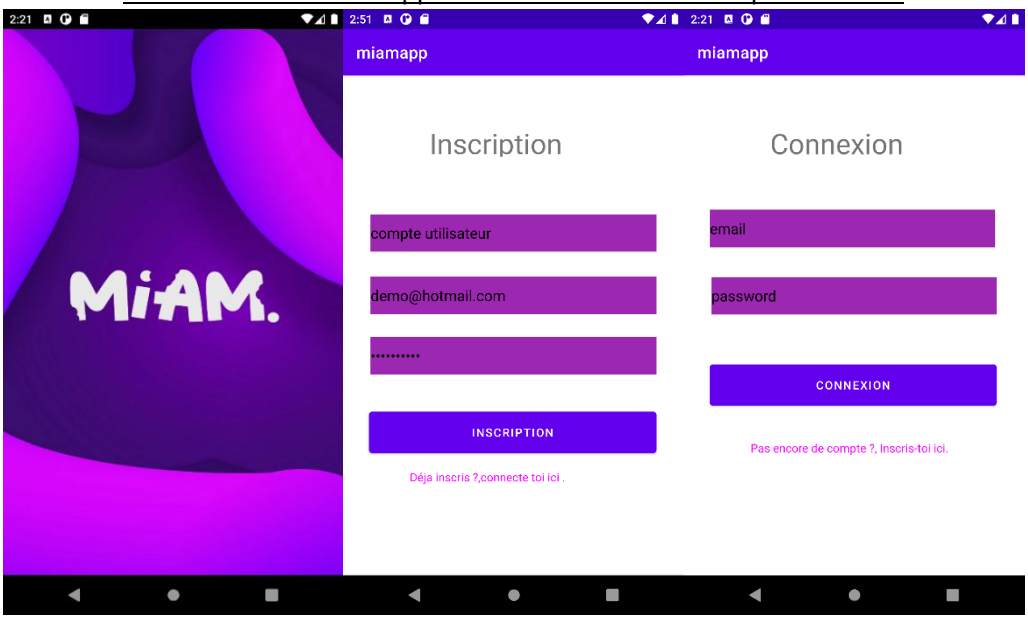
DESCRIPTION DE L'ÉTAT ACTUEL DU PROJET :

Le projet est actuellement terminé à 90 % , toutes les fenêtres ainsi que les interactions complexes ont été gérer, une refonte de la charte graphique a été également faite, toutes les parties de l'application ont été créé, les modifications seront vue en détail et énumérer à la suite de ce paragraphe point par point :

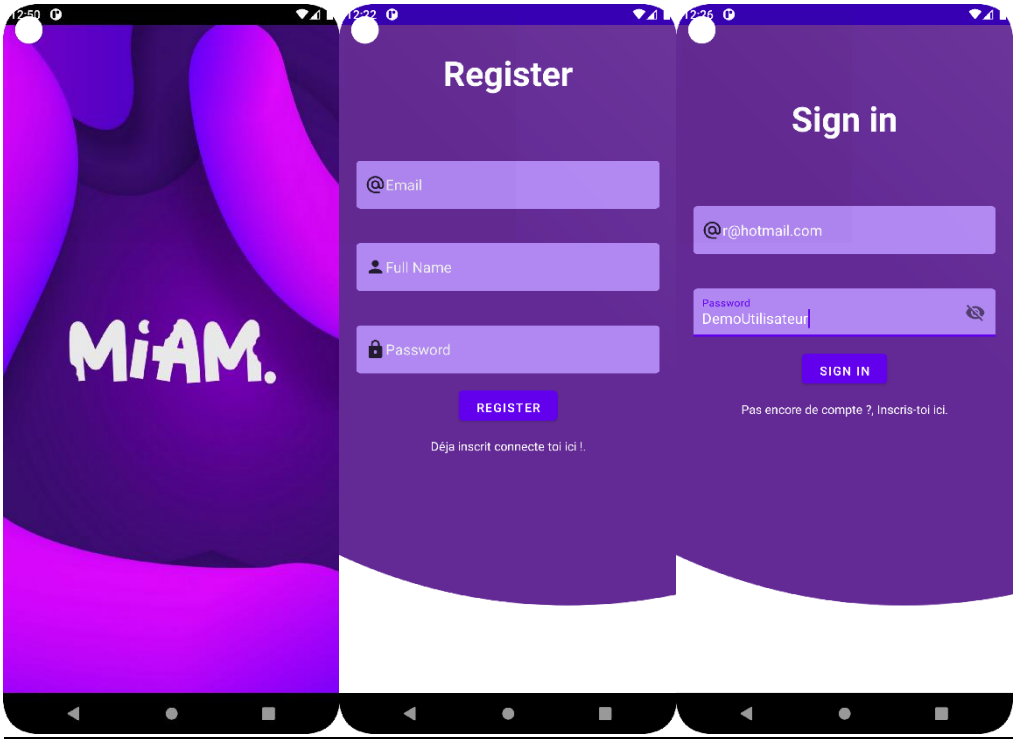
a) Design de l'application actuelle suivis d'explication technique :

Le design a été revu complètement tout en respectant le code couleur voulu, le design a été personnalisé au maximum à l'aide des outils android et l'utilisation d'outils de design externe tel que Photofiltre et Photoshop.

Partie Ouverture de L'application et connexion inscription Avant :



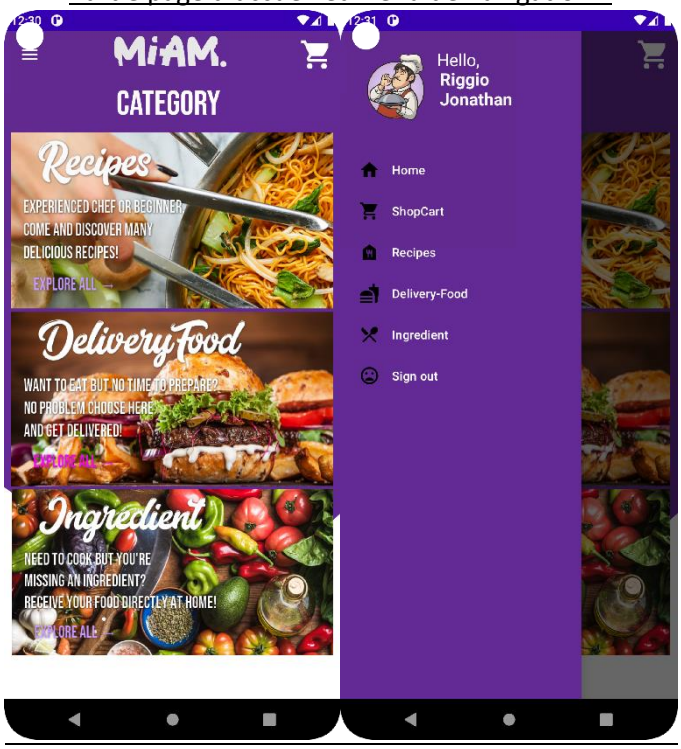
Partie Ouverture de L’application et connexion inscription Aprés :



Explication :

Cette partie comporte toujours l'écran splash, cependant l'écran connexion inscription comporte le même design que toutes les autres parties de l'application ainsi que des éléments maximisant l'expérience utilisateur tel que l'icon œil qui permet d'afficher le mot de passe caché.

Partie page d’accueil et menu de navigation :

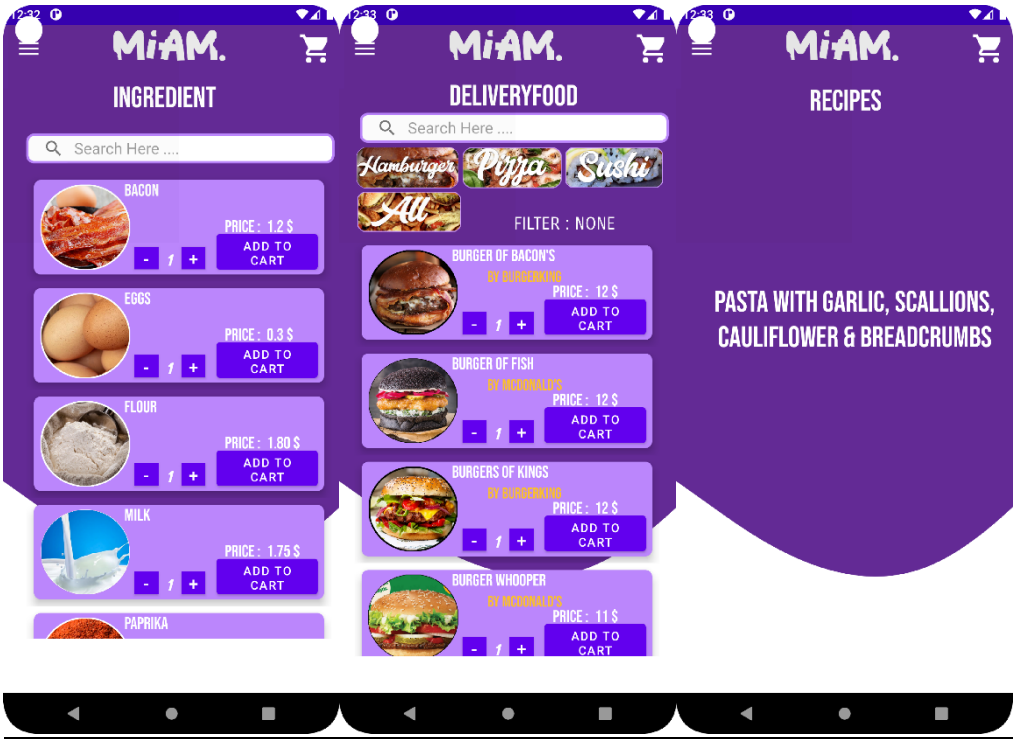


Explication :

Après une inscription réussie voire une connexion, l'utilisateur est redirigé vers la page d'accueil catégorie qui comporte les 3 sections de l'application sous forme d'image bouton créer sur Photofiltre, un menu a été également créer afin de faciliter la navigation entre les pages, il s'affiche lorsque l'utilisateur clique sur l'icon burger, menu qui affiche une photo ainsi que le nom de l'utilisateur afin d'augmenter l'expérience utilisateur.
La barre en haut de la page qui comprend l'icon du menu, le logo de l'application ainsi que l'icon pour le panier d'achat sont présents sur toutes les pages mises à part sur la page payement.
Note : ici si l'utilisateur s'est déjà connecté à l'application, il est directement redirigé vers la page d'accueil sans passer par la page connexion inscription.

Les catégories sont représentées sous forme de bouton imagé avec un slogan accrocheur.

Les 3 parties de l'application :



Explication :

Après avoir cliqué sur une des catégories présentes sur la page d'accueil l'utilisateur est redirigé vers la catégorie souhaitée, celle-ci comporte le même design que le reste de l'application ainsi que la contenue souhaiter.

Ici le contenues de la partie ingrédient ainsi que deliveryfood sont affiché par rapport a la base de données firebase,les ingrédients ainsi que la nourriture fastfood correspond à des éléments dans la base de données, si un élément et ajouter, il s'ajoute automatiquement à l'application ce qui créait une application complètement autonome ne nécessitant pas de modification suite à l'ajout d'un nouvel élément, les items sont affichés grâce à une recycleView et un composant CardView qui vient s'ajouter de manière dynamique par rapport aux éléments de la base de données.

Les boutons + et -, ainsi que add To cart fonctionnent et ajoutent bien dans le panier d'achat, ce que nous verrons dans la suite du rapport.

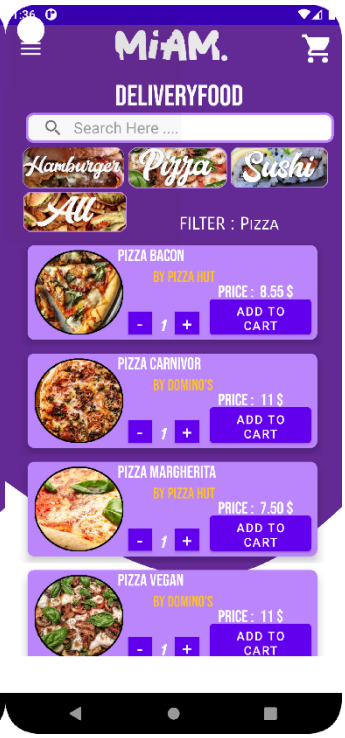
Les fenêtres ingrédients ainsi que deliveryfood contiennent tous les deux une barre de recherche qui facilite l'expérience utilisateur et qui affiche de manière dynamique la contenue souhaiter.
La partie deliveryfood contient également des boutons filtres créer sur Photofiltre qui correspond aux différentes catégories de fastfood ici Hamburger, Pizza, Sushi et pour tout afficher suite à l'application d'un filtre, un bouton All permet de le fair, ici le filtre appliqué est affiché au-dessus de la liste afin de ne pas perdre l'utilisateur.

Exemple :

Recherche ici « ba »

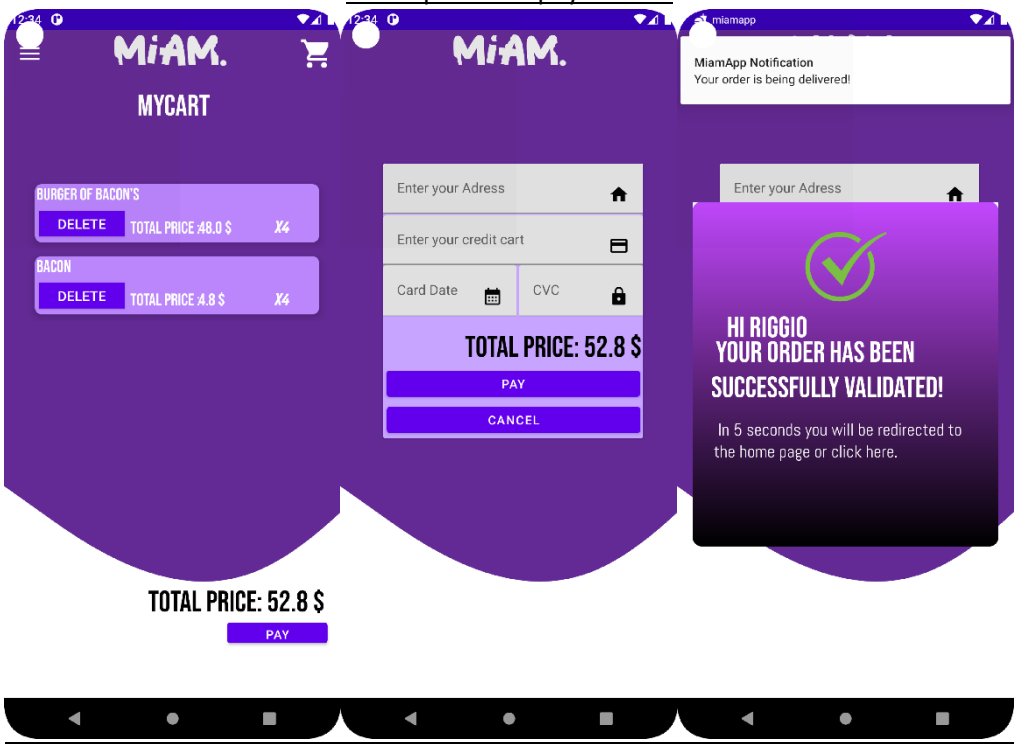
Filtre sushi appliqué

Filtre pizza appliqué



Note : La partie recipes qui correspond à la catégorie recette, n'affiche pas encore tout le contenu souhaiter car dans cette iteration, une focalisation sur la partie complexe a été faite, la partie api et recette a été délibérément délaisser pour se concentrer sur les parties nécessitant beaucoup de recherches.

Partie panier et payement :



Explication :

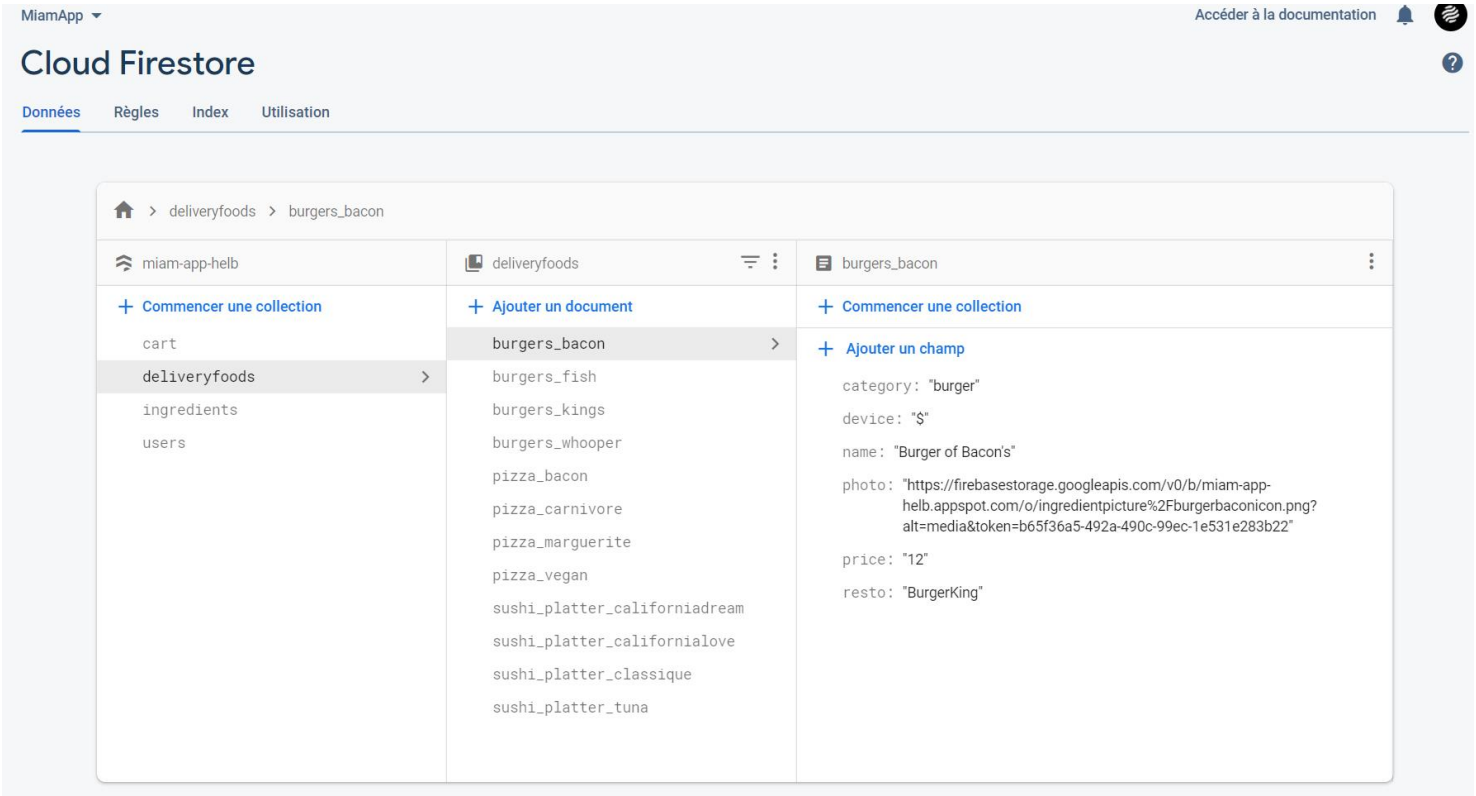
Cette partie traite de la fonctionnalité panier, elle affiche là contenue du panier, elle permet la suppression d'un élément du panier, elle affiche sa quantité qui va permettre d'afficher un prix adapter à la quantité, en bas de page le prix total et affiché, il s'adapte par rapport au contenir du panier et réagis de manière dynamique même après la suppression d'un élément, suivi d'un bouton paye, qui ouvre une fenêtre avec un passage de paramètres entre activités telles que demander dans les exigences du projet, ici le paramètre envoyé est le prix, cette fenêtre simule une entrée de cordonner de paiement pour ensuite proposer de finaliser ou d'annuler, en cas de finalisation, une fenêtre pop-up s'ouvre elle indique une validation de payements suivis d'une notification qui simule une livraison en cours, pour ensuite être redirigé l'utilisateur vers l'écran d'accueil après 5 secondes et si ce n'est pas le cas il peut cliquer sur le texte et il est redirigé.

a) Gestion des données à travers une base de données :

Les données sont stocké dans une base de données firestore plutôt qu'une base de données firebase classique, ce choix s'est justifié par une meilleure structure au sein de la base de données, les données sur firebase database sont stocké sous forme de fichier Json dans un fichier unique contrairement à firestore qui possède une structure composée de collection et de document qui permet d'avoir une structure plus explicite.

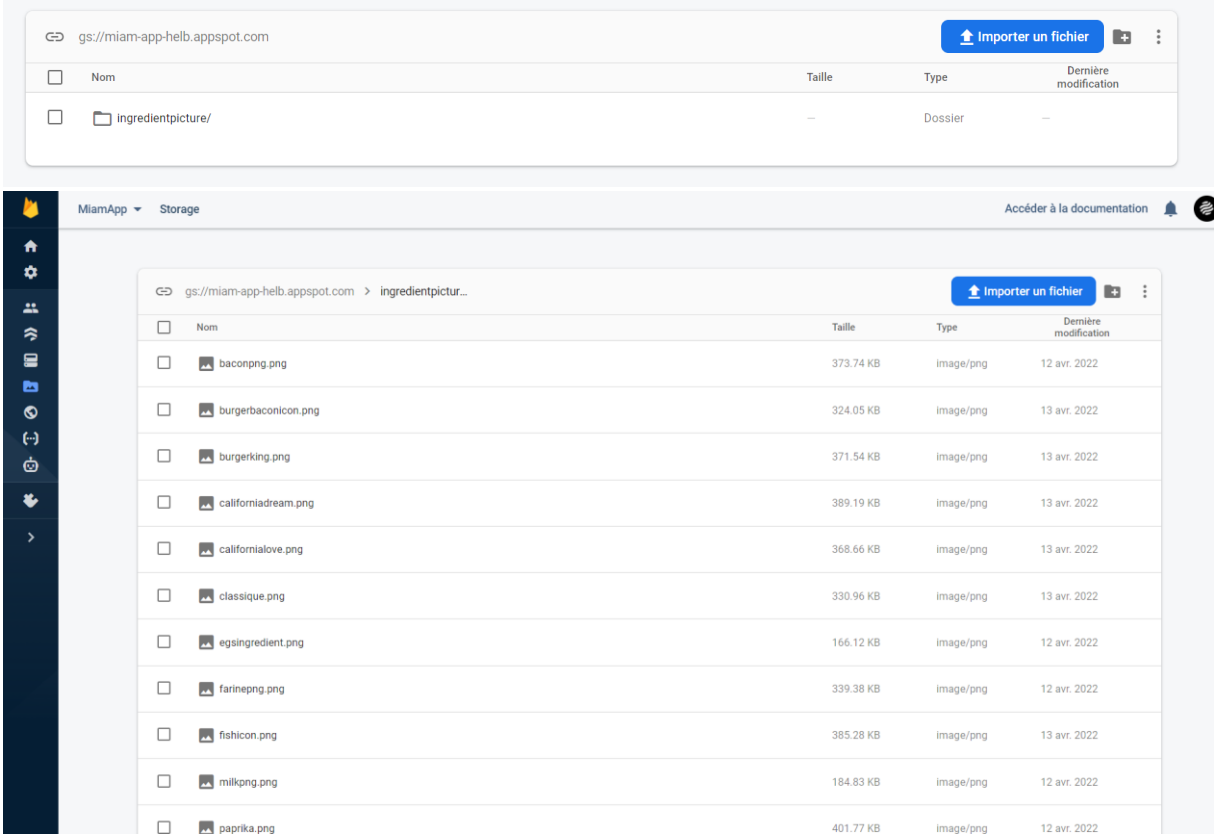
Exemple Ici les ingrédients sont dans la collection ingrédients, les nourritures fastfood sont dans la collection deliveryfoods, etc.

Structure de la base de donnée :



Les images d'ingrédients ainsi que de la nourriture fastfood ont été créer sur Photoshop et stocker dans la partie storage de firebase dans un dossier ingredientpicture/ afin d'être relié avec les items de la base de données grâce à un attribut photo correspondant au jeton d'accès de l'image .

Stockage dans la base de donnée des images d'items :



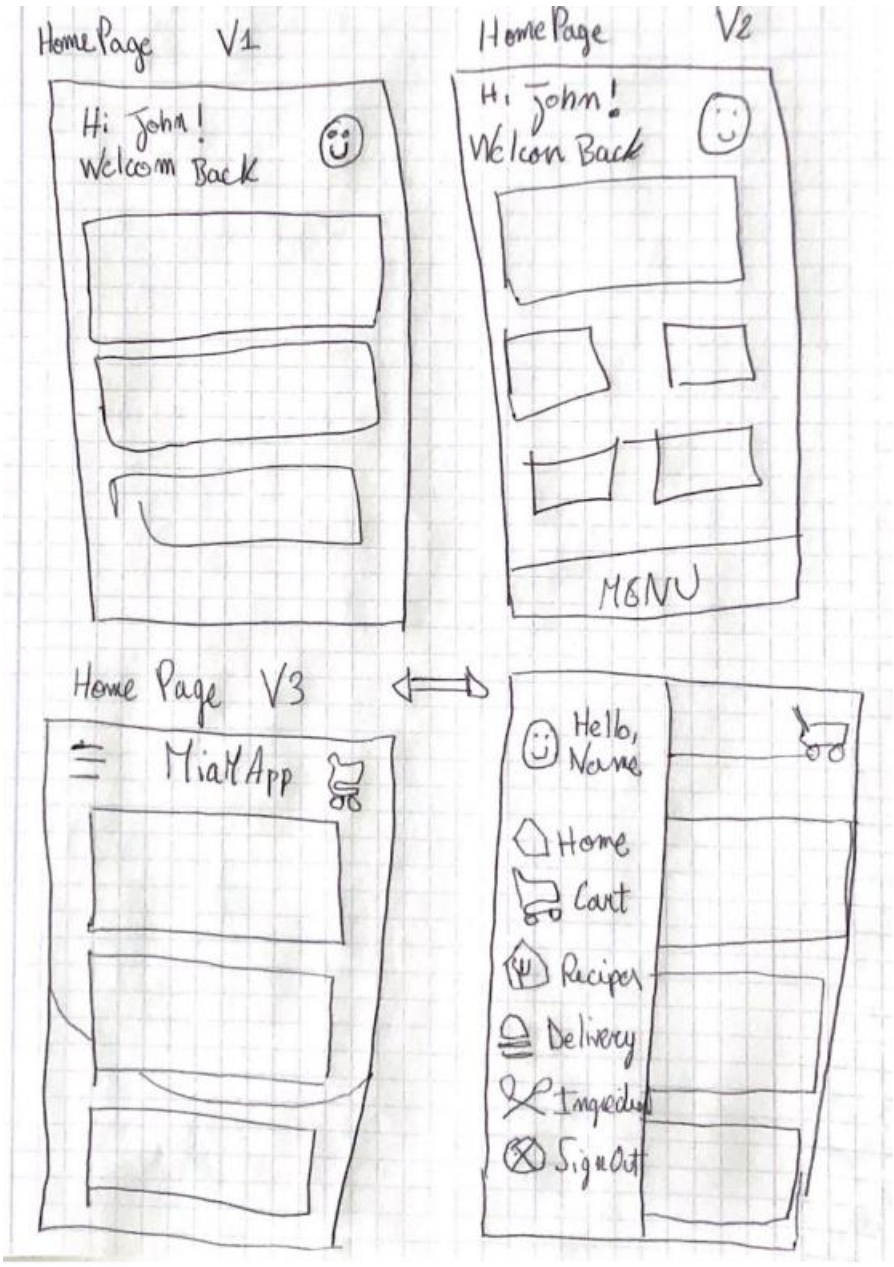
b) Problème rencontré et remarque personnelle :

- La partie design a demandé énormément de temps, une étude des applications déjà existante a été nécessaire pour la recherche graphique ensuite la réalisation de maquette sur papier afin de délimiter les bases graphiques du projet ont permis de déterminer quel type de design pour l'application pour ensuite produire les composants graphiques, certains composants comme les boutons de catégories ont dû être retravaillé.

Avant :

Après :

Maquette papier :



- Ayant choisi une gestion des données sur firestore plutôt que firebase database,une difficulté supplémentaire s'est presenter, celle-ci étant beaucoup moins démocratiser que firebase database ,ce qui complexifie les recherches à son égard car moins de tuto,de documentation sur la base de données, afin d'utiliser cette structure,une utilisation de la documentation ainsi que beaucoup d'essai ont permis une gestion de cet outil.

  Documentation de firestore database ici.

- Étant actuellement sur plus de 5 projets nécessitant chacun un temps conséquent, l'avancement du projet a dû être ralentie par moments.

- L'Émulateur posant énormément de problèmes,une recherche afin de l'optimiser à été nécessaire,le problème d'utilisation de ressource conséquente a été palier par la delocalisation de l'Émulateur sur un autre espace de stockage beaucoup plus libre, un recours à wiper data dans l'onglet gestion de l'Émulateur afin de l'alléger lorsque sa taille devient conséquente et une bonne solution afin de mieux le gérer.
  Plusieurs tests ont été effectué avec différents devices de l'Émulateur, le pixel 4a semble le plus optimal en therme de temps d'exécution.

- L'Émulateur ne s'est plus lancer,la solution a été de supprimer le fichier multiinstance.lock dans le dossier de l'Émulateur ce qui a résolu le problème.

- L'émulateur met plus de 4 minutes à se lancer,résolue en désactivant gradle dans l'onglet Toggle offline mode.

Problème technique avec l'ide avec message dans la console :

- use a compatible library with a minSdk of at most 16 :
  Solution : resolue en changant version de minSdk à 18

- Skipped 33 frames! The application may be doing too much work on its main thread. :
  Solution : resolue avec set a true android:hardwareAccelerated="true"

CONCLUSION :

En conclusion , l'application comporte énormément de fonctionnalité, toute la structure est déjà posée, malgré un temps de travail conséquent avec d'autres projets, l'application à énormément évoluer, la dernière partie qui reste à travailler n'a pas été inclus dans cette itération car une priorité a été fixée sur les parties inconnues, plus complexe et donc une incorporation de celle-ci sera visible à la prochaine remise.

Lien git de l'application : Git