

Rapport Projet RA

*Maxime Bouton, Lucas Despin, Hugo Boutet,
Arthur Colas, Dali Ben Khemis*

Introduction et contexte :

Dans la plupart des supermarchés, des zapettes mises à la disposition des clients. Celles-ci leur permettent de faire leur courses en un temps réduit en scannant eux même leurs articles au fur et à mesure qu'ils les choisissent dans les rayons. Un écran leur permet ensuite d'avoir accès au contenu de leur panier, au prix de chacun des articles ainsi qu'on montant total du panier. Parallèlement à cette technologie, des applications ont été développées pour permettre aux clients de scanner des produits et d'afficher leurs nutri-scores ainsi que certaines autres données. Ces deux technologies permettent ainsi de faciliter les courses des clients et de les aider à manger plus sainement.

Cependant, elles présentent aussi un inconvénient : celui d'empêcher l'utilisateur d'avoir les mains libres. En plus de cela, ce dernier ne peut pas utiliser les deux systèmes en même temps.

Problématique

Notre connaissance de ces solutions ainsi que les inconvénients qu'elle représentent nous ont ainsi conduit à la problématique suivante :

Comment peut-on permettre, grâce à l'utilisation d'une application mobile exécutée sur des lunettes de réalité augmentée, de faire gagner en efficacité à l'utilisateur lors de ses courses, tout en lui garantissant d'avoir les mains libres ? Et si possible, comment y intégrer les données informant l'utilisateur sur la qualité des produits qu'il achète ?

Etude Fonctionnelle (EF)

L'application doit permettre à l'utilisateur d'avoir un suivi en temps réel de son panier. Il doit avoir accès aux produits qu'il contient. Il doit aussi pouvoir le gérer facilement, par exemple en supprimant les articles qu'il ne souhaite plus acheter. Il doit aussi avoir accès aux données disponibles sur le produit, en particulier aux données nutritives. Tout ceci doit être possible via l'utilisation des lunettes, le client ne doit pas avoir besoin d'utiliser ses mains pour effectuer ces opérations.

Modélisation SysML de l'EF

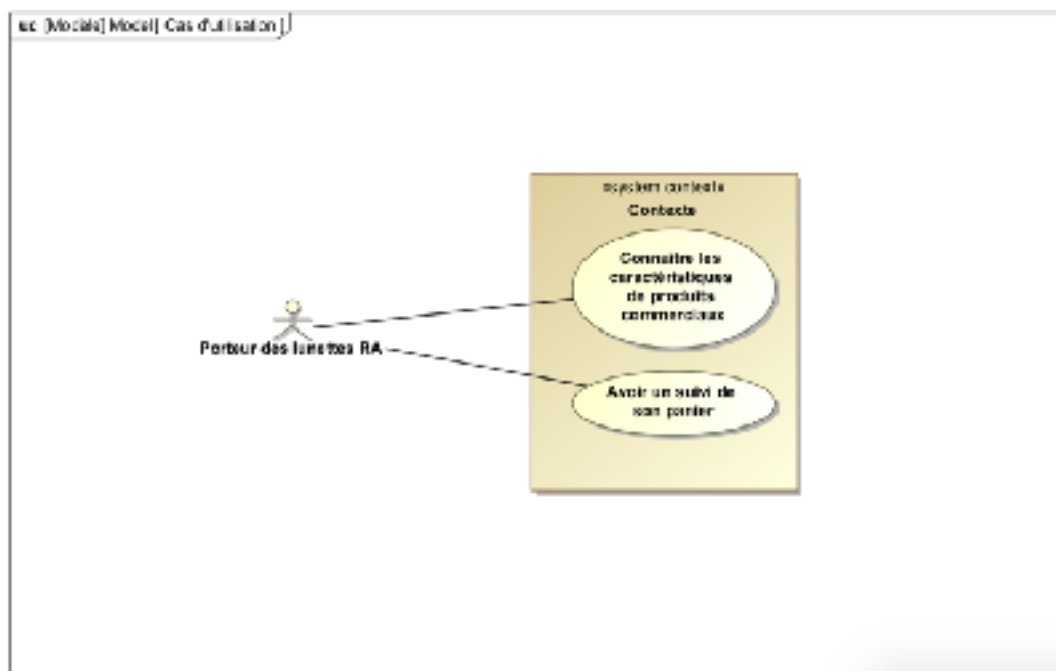


Diagramme des cas d'utilisation

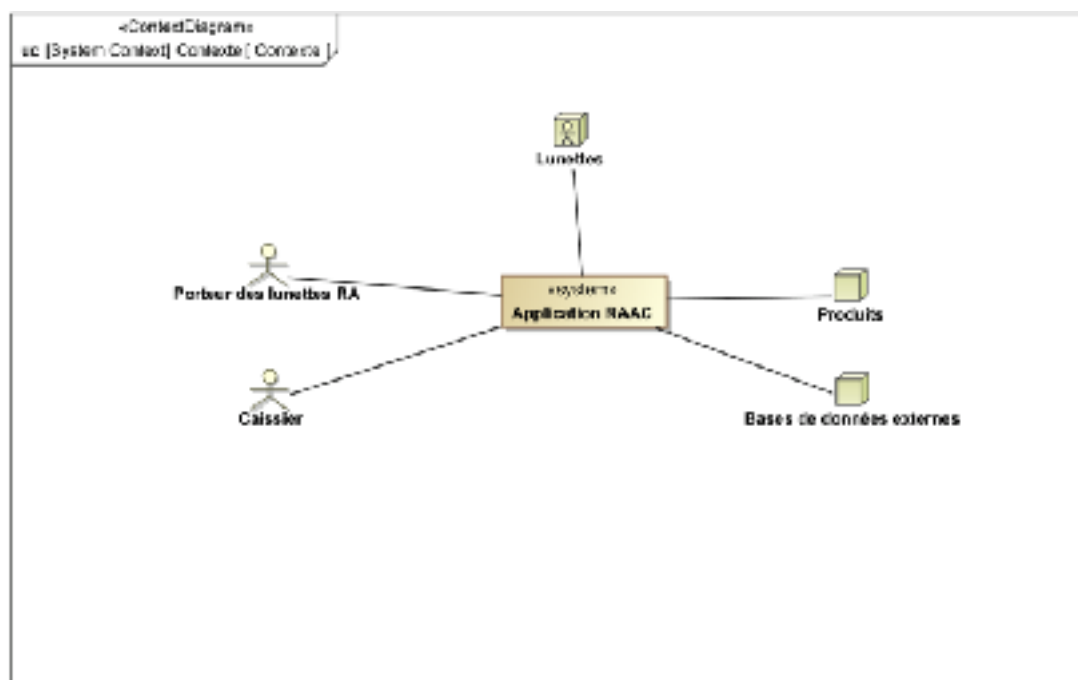
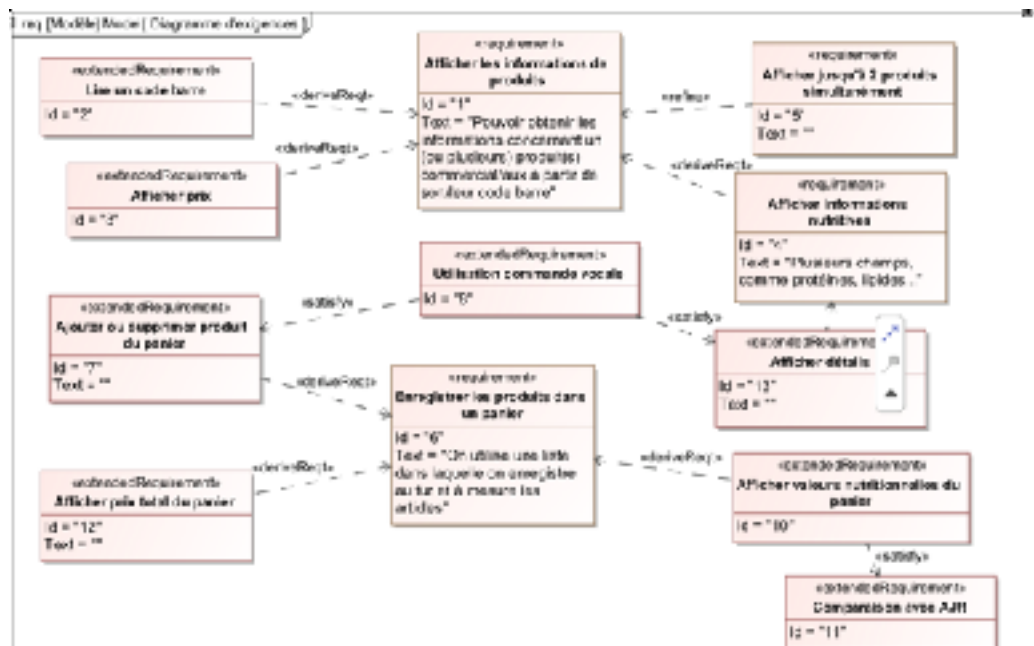


Diagramme de contexte



Diagrammes d'exigences

Outils de développement et langage de programmation utilisés

La totalité de l'application a été développée en Java sous la version Pie d'Android. Nous avons travaillé sur l'environnement Android Studio. Pour récupérer des informations dans une base données en ligne, nous avons aussi dû effectuer des requêtes vers une api en Http.

Architecture informatique de la solution

Pour concevoir une application permettant de répondre à ses attentes, nous nous sommes d'abord concentrés sur le développement d'une application sur smartphone, que l'on pourrait développer et améliorer pour l'utiliser des lunettes de réalité augmentée.

Ainsi, nous avons choisi que notre activité principale permettrait directement de scanner les articles. Cette activité posséderait un bouton permettant d'avoir accès à notre panier, dans une seconde activité.

Enfin, une troisième activité serait lancée lorsqu'on clique sur un produit dans le panier, permettant d'afficher les détails liés à cet article.

Notre MainActivity, renommée ScanActivity, contient donc un Image Button, qui nous permet d'accéder au panier (via un OnClickListener) ainsi qu'un Camera Preview et un TextView indiquant à l'utilisateur de scanner son article. La bibliothèque ZXingScanner nous permet ensuite de détecter un code barre. Une fois ce code détecté, l'article est ajouté dans le panier et le code à 9 chiffres associé à son code barre est récupéré. Ce code est essentiel pour accéder aux

bases de données contenant les informations sur les produits scannés. Ces bases sont en effet open sources, et accessibles via Open Food Facts. En effet, cette association met à la disposition des utilisateurs des fichiers json accessible via une adresse du type suivant : [https://fr.openfoodfacts.org/api/v0/produit/\[code barre\].json](https://fr.openfoodfacts.org/api/v0/produit/[code barre].json)

Lors du clic sur l'ImageButton, l'utilisateur peut alors accéder à son panier. Cette activité se compose d'un TextView qui sert de titre à la page : « Votre panier se compose de ». Elle contient aussi un RecyclerView, qui permet d'afficher la liste des éléments contenus dans le panier. Parmi les solutions existant pour afficher des listes d'éléments, nous avons choisi cette dernière, qui permet de ne charger qu'un nombre réduit d'élément à la fois, ce qui évite à l'application de crasher lorsque l'on manipule de grandes listes.

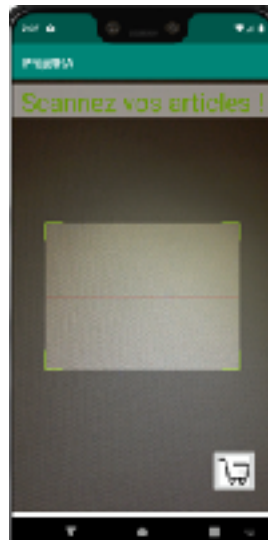
Pour créer cette activité, nous avons dû créer une classe Articles qui nous permet de créer des articles en tant qu'objets. Chaque article est défini par un nom, une description et un code qui permet d'accéder, via des requêtes http, à ses caractéristiques. De même, nous avons dû créer une classe Panier. Chaque objet panier étant caractérisé par une liste d'articles.

Interactions avec les lunettes

Nous avons ensuite adapté cette application aux lunettes. Pour la partie scan de l'article, rien ne change, la caméra des lunettes est utilisée dans les même conditions que celle du smartphone. Pour améliorer l'application et faciliter son utilisation, nous avons simplement ajouté deux commandes vocales. L'une est exécutable depuis la ScanActivity et permet d'accéder à son panier, l'autre se lance depuis l'activité Détail et permet d'ajouter des articles au panier. Ces commandes sont possibles grâce à la bibliothèque DroidSpeech.

Simulations et résultats obtenus

Au lancement de l'application, on obtient bien la vue de la caméra qui cherche à scanner un code barre :



La zone dans laquelle le code barre doit être positionné est bien mise en évidence et, dans la coin inférieur droit, un bouton nous permet d'accéder à notre panier. Après le scan d'un article, celui-ci s'ajoute directement au panier. On indique alors à la caméra de se bloquer pendant 3 secondes, le temps que le client puisse reposer l'article. Ceci permet d'éviter de scanner des produits involontairement.



Lecture code barre

Après un clic sur le bouton panier, ou prononciation de la commande vocale « panier », l'utilisateur a bien accès à la liste de ses articles. Il a aussi la possibilité de supprimer chacun d'eux en cliquant sur la croix située à droite du nom du produit.



Affichage du panier

Une fois dans l'activité Panier, si l'on clique sur le nom d'un produit, l'activité de détail de l'article est lancée. On a ainsi accès à ses ingrédients, son nom, son origine, ses labels, son indice nutritionnel, sa marque, sa quantité ainsi que les catégories auxquelles il appartient.



Affichage des détails

Objectifs atteints

On a bien réussi à développer une application qui permet à l'utilisateur de scanner des articles tout en ayant les mains libres pour l'ajouter à son panier. Le client peut aussi accéder à son panier par une simple commande vocale.

En utilisant le pavé tactile des lunettes, il peut aussi facilement supprimer des articles de son panier et accéder à leurs détails.

Objectifs non atteints

Nous ne sommes cependant pas parvenus à mettre en place une commande vocale permettant d'afficher les détails des produits ou de supprimer des articles depuis le panier. Leur nombre pouvant être élevé, il était compliqué de déterminer par une telle commande sur lequel des articles on souhaitait agir.

De plus, nous n'avons pas pu intégrer le prix dans les données de l'article. Ce dernier dépend en effet du distributeur et n'est pas accessible directement depuis la base de données Food Facts.

Difficultés rencontrées

La première difficulté rencontrée a été de traiter le fichier JSON fourni par la base de données. Il a fallu l'interpréter et la comprendre avant d'en retirer les données intéressantes pour notre application. Ces difficultés résidaient principalement dans le fait que le JSON fourni par Food Fact semblait dépendre des articles scannés.

Selon le code barre reconnu, le nom des clés pouvait varier. De ce fait, il se peut que notre application ne soit pas totalement « universelle ».

Conclusion

Nous avons ainsi pu développer une application fonctionnelle répondant globalement à notre cahier des charges. Le temps imparti ne nous a pas permis d'y répondre parfaitement et certains points restent améliorables. Nous avons toutefois conçu une application permettant à un client de gérer ses courses efficacement, facilement et sans nécessité d'utiliser ses mains pour la plupart des fonctionnalités. Les améliorations peuvent alors avoir lieu au niveau de la création de nouvelles commandes vocales permettant une utilisation totalement « hands free », notamment dans la suppression d'article dans le panier ou dans la demande d'affichage de détails. Des améliorations peuvent aussi être mises en place au niveau du design de l'application.

Finalement, un ajout intéressant serait la possibilité d'accéder au prix des différents articles.

Annexes A (manuel d'utilisation)

Au lancement de l'application, vous pouvez directement scanner le code barre d'un article. Ceci vous permet de l'ajouter directement à votre panier. Depuis cette interface de démarrage, vous pouvez aussi cliquer sur le bouton Panier, ou prononcer la commande vocale « panier » pour en afficher le contenu. Une fois dans votre panier, vous pouvez cliquer sur la croix à droit d'un article pour le supprimer. Vous pouvez aussi cliquer sur le nom d'un article pour en afficher le détail. Vous aurez ainsi accès à différentes informations sur le produit (son nom, son indice nutritionnel, ses ingrédients...).

Annexes B (manuel de développement)