



Application Mobile

ChowDetails



ECAM

Brussels Engineering School

Degrelle Christophe
El Alami Ismaël
El Mourabit Abdoullah
El Jebari Yassine
Taymans Rémy

Table des matières

- 1. Présentation 2
- 2. Activités 3
 - 2.1. FoodMainActivity 3
 - 2.2 FoodSearchActivity..... 4
 - 2.3. FoodDetailActivity 4
 - 2.4. FoodFavoriteActivity 5
 - 2.5. FoodPreferenceActivity 5
- 3. Code barre 6
- 4. Remarques..... 6
- 5. Améliorations..... 6

1. Présentation

[GitHub](https://github.com/MrNo-Body/ecam-app-android) (<https://github.com/MrNo-Body/ecam-app-android>)

Notre application, ChowDetail, a pour but de fournir à l'utilisateur toutes les informations concernant des aliments en ventes dans tous les pays.

Elle ne se limite pas à de la simple recherche, chaque produit consultable contient des informations détaillées sur le produit en question : les ingrédients, les allergènes, les additifs alimentaires, la marque du produit, etc. L'utilisateur fait une recherche par nom ou scan un code barre via l'appareil photo.

Finalement, cette application permet d'une part d'enregistrer en favoris les produits souhaités pour avoir un accès rapide et hors connexion des produits fréquemment consultés, et d'autre part de modifier certains paramètres propres à l'application tels que mettre un mode nuit entre autres.

Cet outil offre un service non négligeable quand on constate les quantités de produits vendus dont les informations sont parfois insuffisantes voire inexistantes.

Pour avoir toutes ces informations, nous avons repris les données d'une API, OPEN FOOD FACTS (<https://en.wiki.openfoodfacts.org/API>).

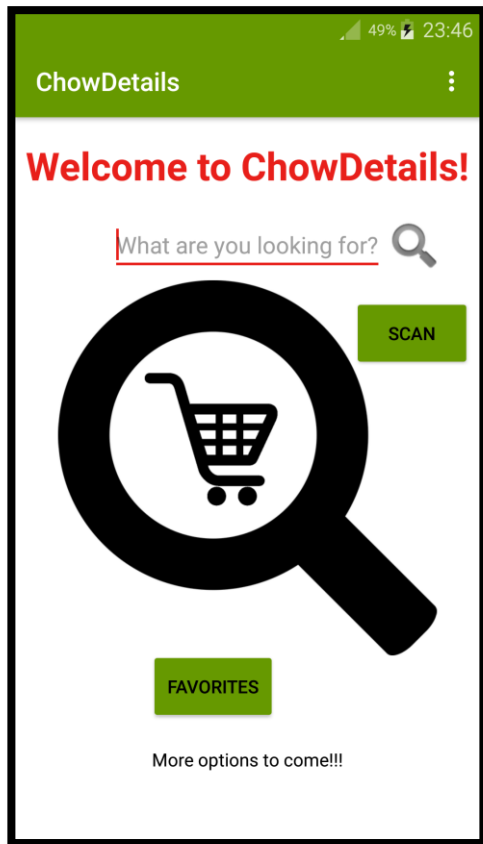
Du point de vue logiciel, l'application comprend 5 activités :

- FoodMainActivity
- FoodSearchActivity
- FoodDetailActivity
- FoodFavoriteActivity
- FoodPreferenceActivity

Détaillons donc maintenant leurs fonctionnements une par une.

2. Activités

2.1. FoodMainActivity

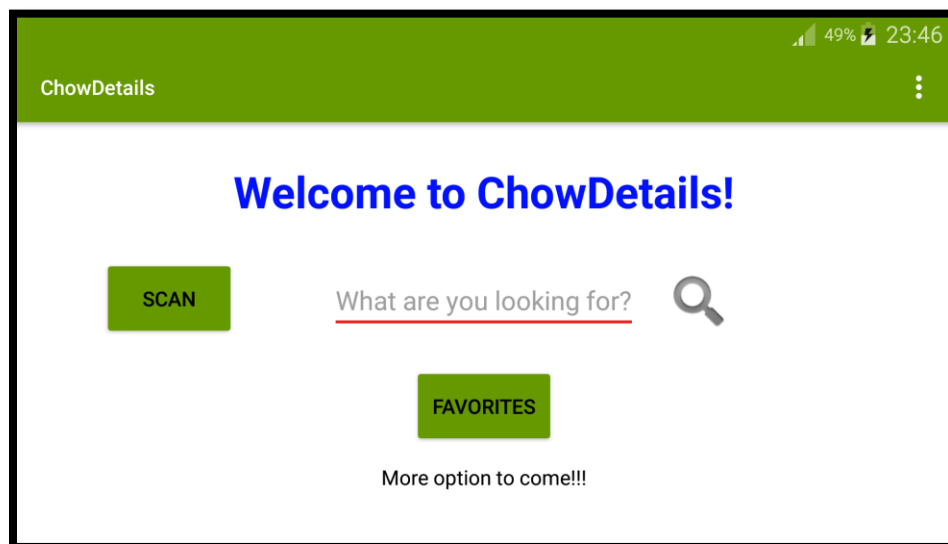


Le page d'accueil permet d'entrer le produit recherché, si l'utilisateur appuie sur le bouton "FIND", il sera alors renvoyé vers l'activité "FoodSerachActivity", tandis que s'il appuie sur le bouton "Accéder aux favoris", il sera envoyé vers l'activité "FoodFavoriteActivity".

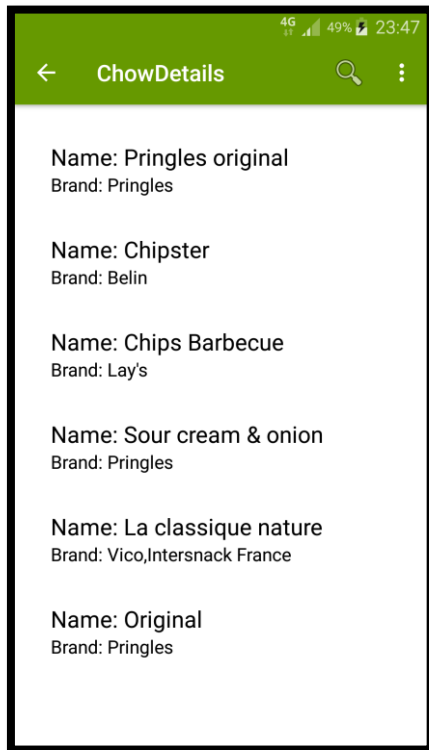
Le bouton scan permet de récupérer le code barre d'un produit grâce à l'appareil photo.

Il peut aussi à tout moment appuyer sur le bouton en haut à droite pour accéder aux préférences via l'activité "FoodPreferenceActivity".

FoodMainActivity utilise le layout activity_food_main.xml en mode portrait et activity_food_main.xml (land) pour le mode paysage.



2.2 FoodSearchActivity



Ici, le layout `activity_food_search` est utilisé.

L'objectif de cette activité est de :

- Construire l'URL à envoyé afin de récupérer un json
- Récupérer les résultats (Nom des produits)
- Utiliser un `ItemAdaptater(RecyclerView)` afin d'afficher correctement les résultats de la recherche.

L'URL est constituée comme décrit dans les informations d'OPEN FOOD FACTS. (Explication d'envoi de requête : https://en.wiki.openfoodfacts.org/API#Searching_for_products)

La classe `NetworkUtils` est utilisée pour faire notre requête et la classe `ItemAdaptater` pour afficher les résultats.

2.3. FoodDetailActivity



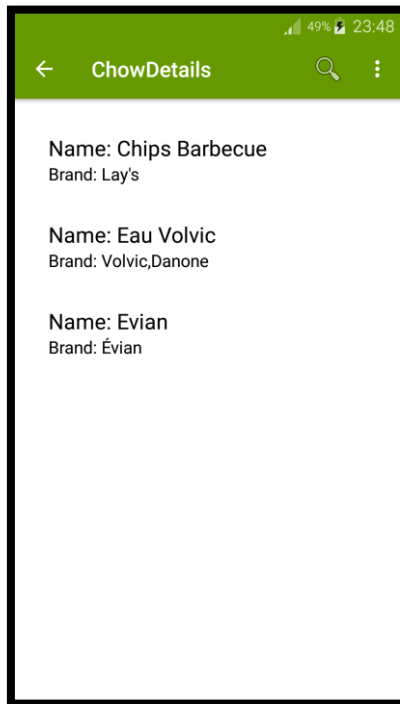
Cette activité affiche le détail d'un produit. Comme son nom, les catégories dans lesquels il est rangé. Il est possible d'obtenir plein d'informations concernant un produit. Toutes les informations ne sont pas disponibles dans l'application pour l'instant mais tout est en place pour facilement aller rechercher ces informations sur le serveur de donnée. Il faut juste ajouter des `TextView` supplémentaires.

C'est dans cette activité que l'on peut choisir de marquer un produit comme favori ou de le retirer

L'interface est une `ScrollView` qui permet ds'afficher toutes les informations, elle se trouve dans le fichier `activity_food_details.xml`.

On arrive dans cette activité à partir de l'activité `FoodSearchActivity` ou `FoodFavoriteActivity`.

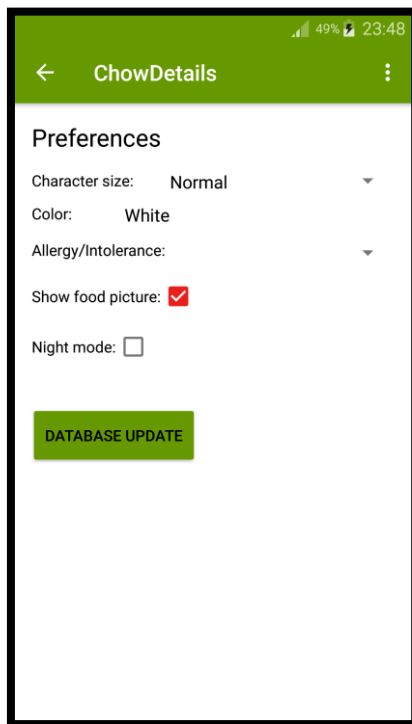
2.4. FoodFavoriteActivity



Cette activité présente la liste des produits marqués comme favori par l'utilisateur. Elle est similaire à l'activité FoodSearchActivity. Elle utilise une RecyclerView comme FoodSearchActivity et le même layout pour afficher les éléments de la liste.

La différence avec FoodSearchActivity est que FoodSearch affiche une liste après avoir fait une recherche sur l'API en ligne. Tandis que FoodFavoriteActivity va chercher les produits qui sont enregistrés dans la base de données locale.

2.5. FoodPreferenceActivity



Le but de cette partie est de permettre à l'utilisateur de modifier les paramètres de son application. Il peut changer la taille de l'écriture, activer un mode nuit, décider d'afficher ou non les images. Une autre fonction est de pouvoir filtrer les aliments suivant leurs allergènes, mais elle n'est pas complètement opérationnelle.

Le thème peut être changé via le mode nuit ou l'option couleur. Le bouton "mettre à jour la base de données" sert à mettre à jour la base de données avec quelques termes prédéfinis.

La gestion de couleur est implémentée avec une défensive qui évite le crash de l'application.

3. Code barre

Utilisation d'une librairie open source(ZXingScanner) pour pouvoir scanner le code barre, malheureusement la fonctionnalité n'est pas pleinement opérationnelle.

4. Remarques

La flèche de retour dans la menu option renvoie toujours vers la page d'accueil, afin de mettre à jour toutes les vues. Ceci est un choix de conception, il peut être optimisé en utilisant à la place un intent avec retour tel qu'utilisé dans la page des détails avec le retour vers la page des favoris.

La touche de retour de la page des détails fonctionne différemment en fonction de la vue précédente. Celle qui renvoi à la vue recherche est un simple retour, tandis que celle qui renvoi à favoris est un intent avec retour qui initialise le processus de "ondestroy()" et "oncreate()" grâce à "startActivityForResult()" dans l'activité des favoris.

On a choisi d'utiliser une API plus ancienne pour augmenter la compatibilité entre téléphone. Ce qui nous empêche d'utiliser certaine fonctionnalité plus optimisée.

Le diagramme de classe est en annexe sur [GitHub](#).

Origine du nom: jeux de mot avec Chow (=bouffe) et Show (=montrer), le nom peut se comprendre soit monter les détails ou détails de la bouffe.

5. Améliorations

- Trier alphabétiquement la liste des favoris
- Menu déroulant pour aller sur la page qu'on veut directement
- Permettre d'ajouter des produits
- Ajouter d'autres détails pour les produits
- Afficher l'état de la recherche, afficher si on ne trouve rien
- Signaler quand la connexion internet est interrompu ou non existante
- Ajout d'un toast de confirmation d'ajout au favori
- Utiliser ce qui existe "PreferenceScreen" pour les préférences
- Utiliser une API minimum plus élevé pour utiliser les fonctionnalités de transition et d'interface plus élaborée.
- Mettre à jour les détails des favoris
- Ajouter une option à propos pour avoir des informations sur les développeurs