



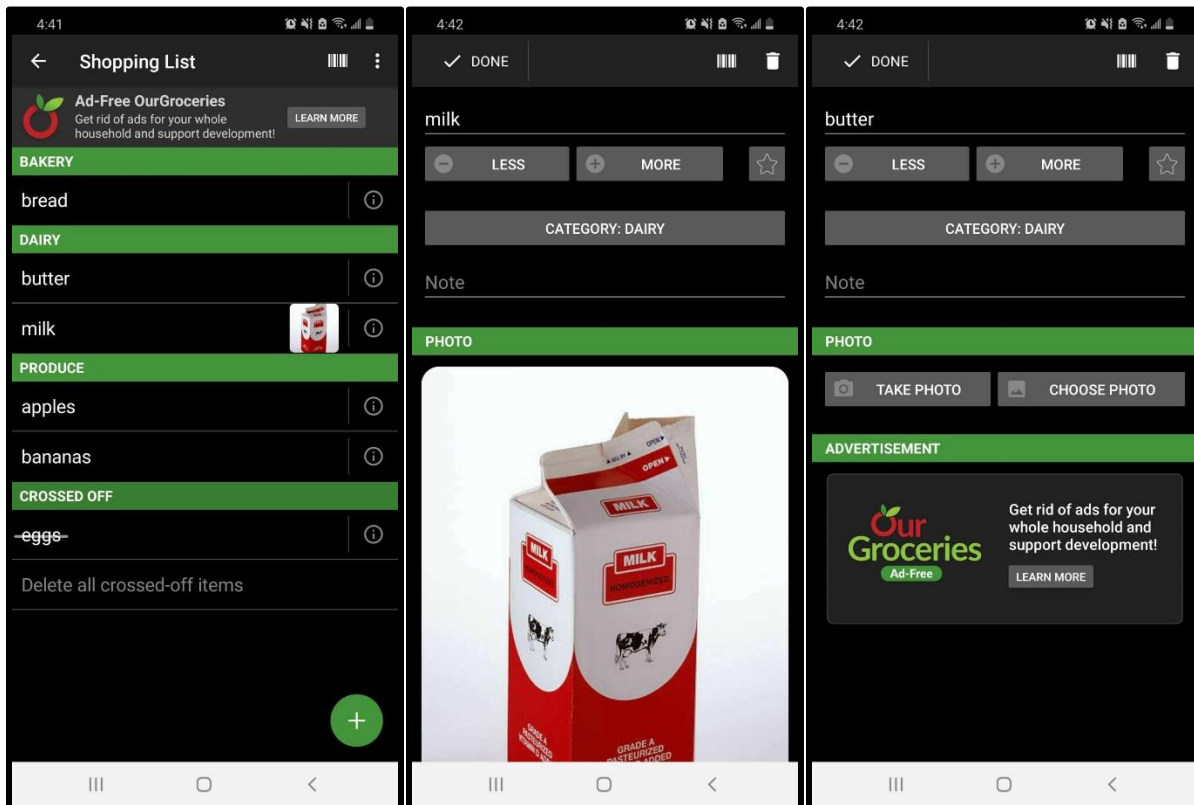
Sujet : Travail Pratique 1 : Liste d'épicerie
Valeur : 30% du semestre
Date de remise : 27 octobre 2024 à 23h59
Mode de remise : remise automatique sur GitHub

Instructions générales

- ❖ L'API 26 (Oreo, Android 8.0) doit être utilisée pour le projet
- ❖ La vue doit fonctionner en **paysage** et **portrait** pour le téléphone **Pixel 3a API 35**
- ❖ Les listes doivent utiliser des **RecyclerView**
- ❖ La BD utilisée doit être locale et utiliser la librairie **Room**
- ❖ Le travail se fera en **équipe de 2**

Contexte

L'objectif de ce travail pratique sera de programmer une application de liste d'épicerie. Vous pouvez vous inspirer des applications déjà existantes comme celle dans les photos, *Our Groceries* ou d'autres comme *Keep* et *Listonic*.



Liste d'épicerie

La première étape du projet serait de faire une liste d'épicerie qui intègre la vue RecyclerView. Chaque item de la liste devrait minimalement afficher le nom, la quantité, la catégorie, une image ainsi qu'un bouton pour le modifier. On doit aussi pouvoir cliquer sur un item pour l'ajouter à une autre liste qui va contenir les items déjà dans notre panier. Un bouton doit permettre de créer un nouvel item qui va être ajouté à la liste.

- ❖ *RecyclerView* avec une liste des **items à acheter** et un autre pour ceux **dans le panier**
- ❖ Chaque item contient **nom, quantité, catégorie, image** et **bouton** pour le modifier
- ❖ Cliquer sur un item de la liste à acheter devrait l'**enlever** de celle-ci
- ❖ Cliquer sur un item de la liste devrait l'**ajouter** à l'autre liste
- ❖ Créer un nouvel item ajouté à la liste

Modification d'un item

Lorsque nous cliquons sur le bouton d'un item dans la liste, nous devrions être redirigés vers une activité qui permet de la modifier ou de la supprimer. Je devrais être capable de modifier son nom, sa quantité, sa catégorie et sa photo. La photo peut être choisie des images dans l'appareil de l'utilisateur ou directement de sa caméra.

- ❖ Rediriger vers une activité qui contient les **bonnes informations** de l'item
- ❖ Capable de modifier son **nom**, sa **quantité**, sa **catégorie** et sa **photo**
- ❖ La photo peut venir des **images** dans l'**appareil** ou de la **caméra**
- ❖ Capable de **supprimer** l'item

Utilisation de la librairie *Room*

La prochaine étape du projet serait d'instaurer une librairie qui permettrait de sauver l'information localement pour ne pas perdre ses modifications. La librairie requise est *Room*. Alors, je devrais être capable de faire des modifications à mes listes, fermer l'application, la rouvrir et ne rien perdre.

- ❖ Capable de faire des modifications qui sont **sauvegardées localement**
- ❖ Ne pas perdre les modifications lors du changement de **portrait à paysage**

Fonctionnalités supplémentaires

La page précédente contient les fonctionnalités les plus importantes, cependant il manque beaucoup de choses pour faire une application complète. Vous allez devoir décider d'ajouter les fonctionnalités qui, selon vous, manquent à l'application. Voici une liste d'idée non exhaustive :

- ❖ Ajouter des items à une liste de favoris
- ❖ Séparer les items de la liste selon leurs catégories
- ❖ Filtrer la liste selon leurs attributs, ascendants et descendants
- ❖ Créer et se connecter à un compte
- ❖ Ajouter du son et des animations
- ❖ Ajouter la localisation en anglais, espagnol, etc.
- ❖ Ajouter un thème de mode sombre
- ❖ Ajouter une page de paramètres (son, animations, localisation, mode sombre, etc.)

Barème

Critères	Points
Fonctionnement du programme <ul style="list-style-type: none"> Afficher les listes de produit à acheter et dans le panier Ajouter, modifier et supprimer un item Changer la photo d'un item à partir de la caméra ou du cellulaire Sauvegarder l'information localement avec Room 	/60
Fonctionnalités supplémentaires <ul style="list-style-type: none"> Ajout de fonctionnalités supplémentaires Fonctionnement et homogénéité des fonctionnalités 	/20
Ergonomie / qualité de l'interface <ul style="list-style-type: none"> Application utilisable en portrait et paysage Facilité d'utilisation et aspect visuel Respect des normes d'ergonomie d'Android Gérer correctement les erreurs avec des Toasts 	/10
Programmation <ul style="list-style-type: none"> Clarté du code (ajout de commentaire ou nécessaire) Respect des normes de programmation présentées Nom de classe, fonctions et variables claires Éviter la duplication de code et le franglais Utilisation des ressources 	/10
Total	/100