




Guide du développeur

ListEat est une application qui permet à l'utilisateur de gérer ses achats grâce à la création de listes de courses.


Page d'accueil

La page d'accueil est gérée par la classe *MainActivity*. Elle comporte plusieurs fonctionnalités :


- l'affichage des listes créées est géré par la fonction *onCreate* et la classe *MyDBHandler* qui est une base de données de listes et qui permet les opérations de stockage des listes avec, entre autres : ajout, suppression, application des modifications et requêtes.
- l'ajout d'une nouvelle liste grâce au bouton  qui appelle la fonction *addOrEditList* et redirige l'utilisateur vers la page de création de liste. Cette dernière est gérée par la classe *AddOrEditListActivity*.
- la visualisation d'une liste déclenchée par un clic sur le nom d'une liste. Elle sollicite la fonction *displayContent* qui redirige l'utilisateur vers la page détaillée de la liste. Cette page est gérée par la classe *DisplayListActivity*.
- la déletion d'une liste grâce à un long clic géré par la classe *ExampleDialog*.

Chacune des classes introduites précédemment vont être explicitées dans la suite du guide.

Visualisation d'une liste

La visualisation d'une liste est gérée par la classe *DisplayListActivity*. Elle permet également de servir de support à l'utilisateur lorsqu'il part en courses grâce à l'onglet  en particulier. En cliquant sur un élément de la liste, ce dernier est déplacé dans le caddie. A l'inverse, lorsque l'utilisateur clique sur un élément du caddie, il est rajouté à la liste. Pour cela, deux fonctions ont été implémentées pour passer de la vue de la liste à celle du caddie et inversement : *showList* et *showCaddie*. Leur contenu est géré par la fonction *onCreate*.

Lorsque des éléments sont ajoutés au caddie, l'utilisateur doit enregistrer la liste pour que ces éléments disparaissent définitivement celle-ci. Si l'activité est quittée sans que l'utilisateur ait enregistré la liste, alors tous les éléments qui avaient été placé dans le caddie se retrouveront dans la liste.

Cette page permet également à l'utilisateur d'éditer la liste grâce au bouton . Ce bouton renvoie vers l'activité *addOrEditListActivity*. L'ajout et l'édition de liste fonctionnent de la même manière, à la différence que pour l'édition, la page est pré-remplie avec les données de la liste, tandis que pour l'ajout, la page est vide.

Édition et création d'une liste

Comme précisé précédemment, l'édition et la création de liste sont gérées de la même manière, donc par la même classe. Elle impose à l'utilisateur que la liste ait un titre et contienne au moins un élément. Ces éléments peuvent être recherchés dans la barre de recherche grâce à l'API *spoonacular*. Si l'élément recherché n'est pas retourné par la requête à l'API, il peut toujours être ajouté à la liste. Cela permet également de compléter la liste hors connexion internet, puisque l'utilisation de l'API en requiert une.

Répartition du travail

Sag Özgür : Implémentation des classes *AddOrEditListActivity*, *MainActivity*, *MyDBHandler* et *MyList* (objet pour gérer les listes de courses).

Douare Juliette : Partie *front-end* avec la création du logo et des icônes et l'écriture des *layouts*. Implémentation des classes *DisplayListActivity* et *ExampleDialog*.

Pour plus de détails concernant les fonctions et leurs paramètres, se référer au code commenté.