Documentation technique

Projet de Technologies pour Applications Connectées

GENART Valentin

Master Informatique – Université de Lille

2020 – 2021

---

1. Description du projet

Le projet consiste en la création d'une application mobile pour systèmes Android. Cette application permet à un utilisateur de consulter une liste de différentes bières existant dans le monde. Cette liste de bières est récupérée via une API. Une barre de chargement s'affiche au lancement de l'application, le temps que les données soient récupérées et affichées. Une fois la liste des bières récupérées, elle est stockée dans une base de données locale, ce qui signifie que lorsque l'utilisateur quitte l'application et la relance, la liste est directement disponible sans avoir à requêter l'API. La liste affiche, pour chaque bière, son nom, son image si elle existe, sa catégorie et son taux d'alcool. S'il n'y a pas d'image pour une bière dans l'API, une image par défaut est utilisée. Si la catégorie n'est pas disponible dans l'API, le champ correspondant est laissé vide. Une barre de recherche en haut de l'écran permet de rechercher une ou des bières dans cette liste.

Figure 1: Liste des bières

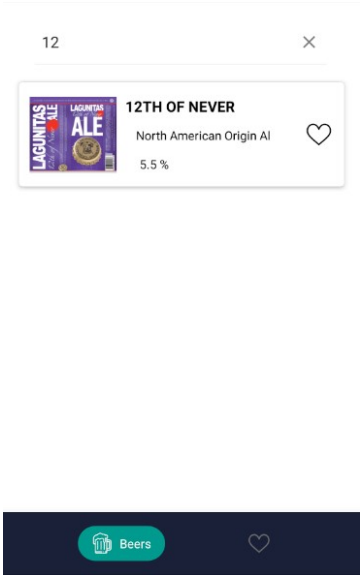

Figure 2: Recherche de bière

La rotation de l’écran change l’affichage de la liste en grille à deux colonnes.

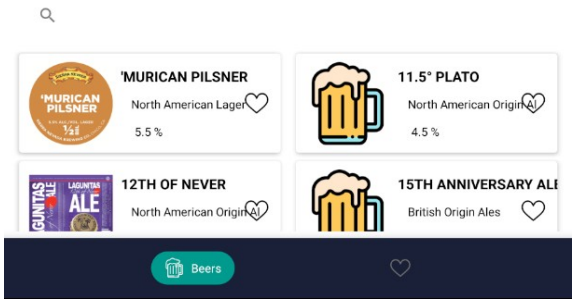

Figure 3: Affichage en mode grille

Le clic sur une bière permet d’accéder à un écran affichant le détail de cette bière. L’image s’affiche en haut de l’écran. Lors du scroll, l’image disparaît en fondu et est remplacée par une barre contenant le nom de la bière. L’écran de détail fournit en plus une courte description, la date de mise à jour de l’API pour cette bière, le degré d’alcool de la bière, sa description, et un statut indiquant que les données de cette bière sont vérifiées.

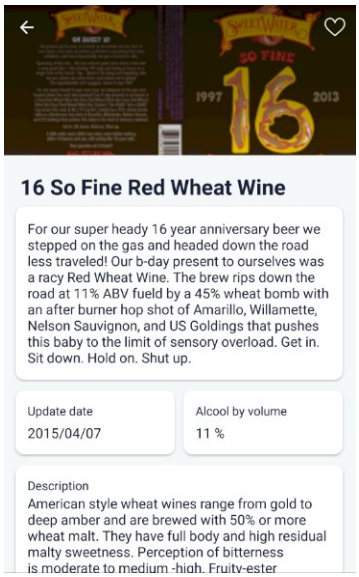

Figure 4: Détail d'une bière

Le clic sur l’icône en forme de cœur, que ce soit sur l’écran affichant la liste ou sur celui affichant le détail, permet d’ajouter la bière en favori. Une barre de navigation en bas de l’écran permet de switcher entrer l’écran affichant la liste des bières et l’écran de favoris. L’écran de favoris contient la liste des bières ajoutées en favori. Pour chaque bière, on retrouve son nom, sa catégorie, son taux

d’alcool et sa description. Le swipe d’une bière de la liste vers la gauche ou la droite permet de supprimer cette bière de la liste des favoris. Cette liste s’affiche également en mode grille à deux colonnes lors de la rotation de l’écran. Tout comme la liste complète des bières, la liste des favoris est sauvegardée dans une base de donnée locale. Ainsi, l’utilisateur pourra la retrouver lorsqu’il relance l’application.

2. Diagramme de séquences

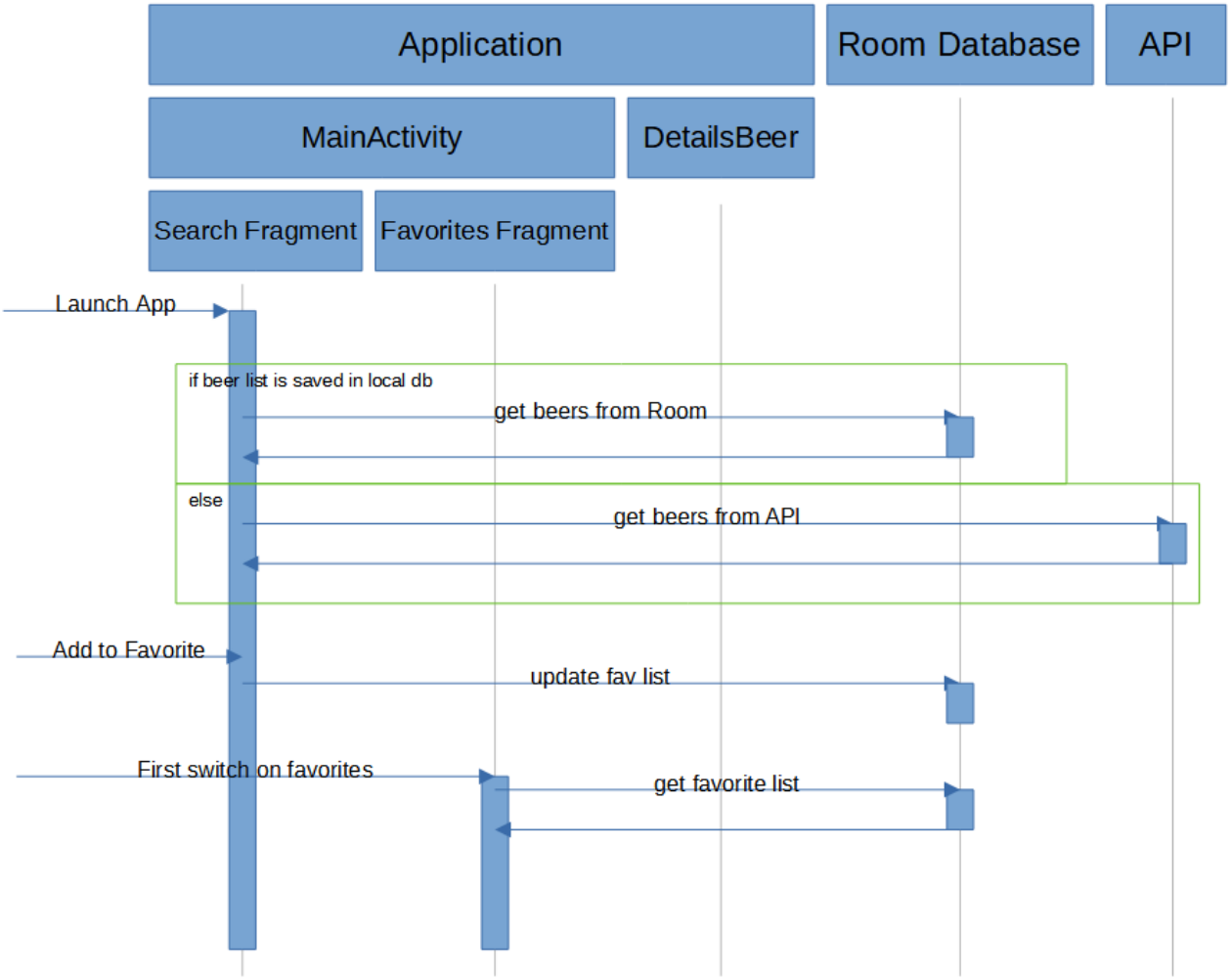

Figure 5: Diagramme de séquences

Ce diagramme montre comment se fait la récupération et la sauvegarde des données. Au lancement de l’application, si la liste des bières est sauvegardée dans la base de donnée locale Room, on récupère cette liste, sinon on appelle l’API pour la récupérer. Lorsque l’utilisateur ajoute une bière en favori, la liste des bières favorites est mise à jour dans Room. Lorsque l’utilisateur switch pour la

première fois sur le fragment favori, la liste des favoris est construite en récupérant la liste des favoris depuis Room.

3. Diagramme de classes

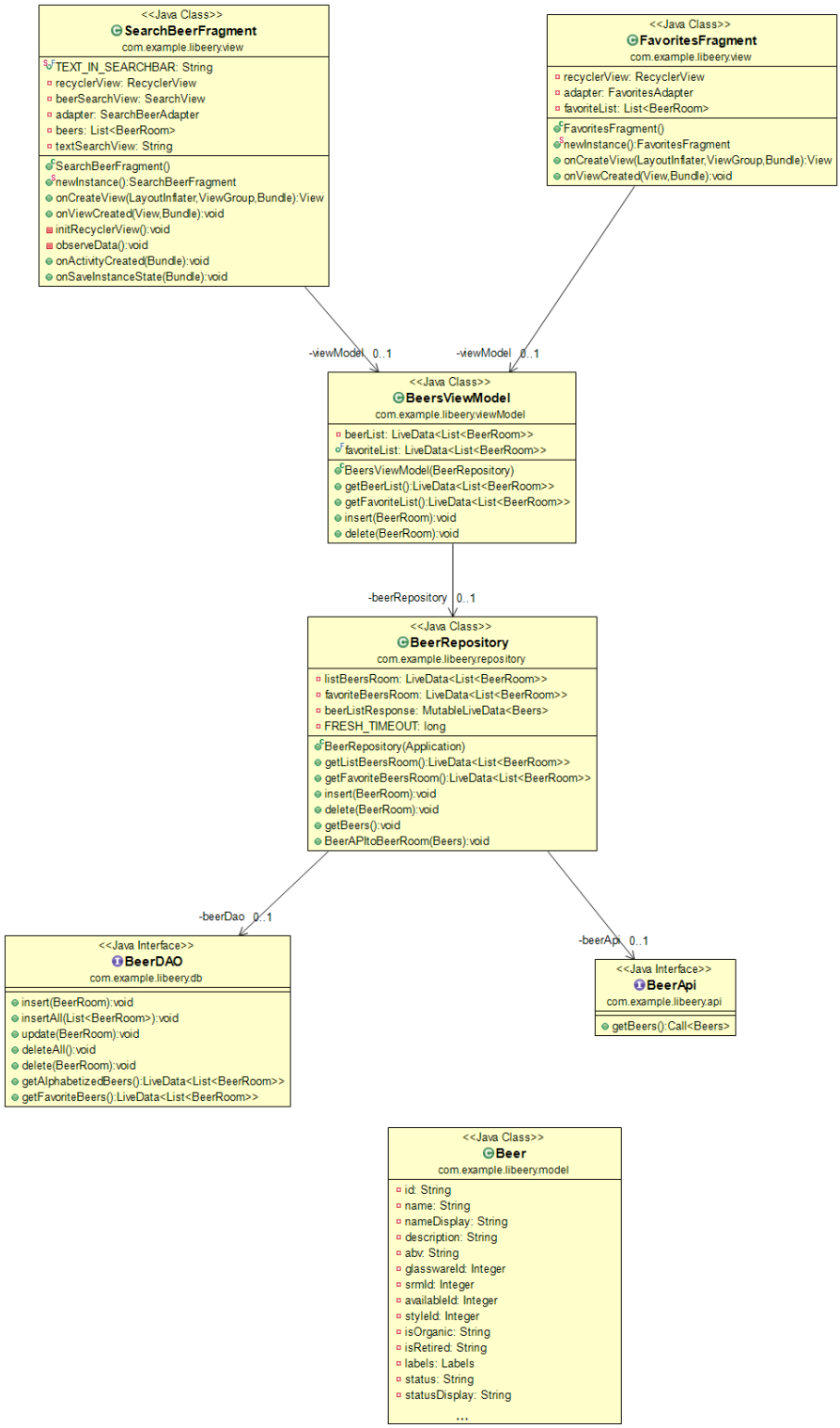

Figure 6: Diagramme de classes UML

4. Contrat d’interface

L’obtention de la liste de bières depuis l’API se fait en appelant l’URL https://api.brewerydb.com/v2/beers?key={key} en type GET. C’est la seule requête que nous avons à faire. Le filtrage des bières par la barre de recherche se fait sur les données du RecyclerView et non en rappelant l’API. Le paramètre « key » correspond à notre clé d’accès développeur à l’API. Il est possible d’utiliser l’URL https://api.brewerydb.com/v2/search?q={query}&type={type} pour obtenir une liste de bières dont le nom contient {query} et où type = « beers ». Nous utilisions cette URL à l’origine mais n’en avons finalement plus l’utilité. Le résultat attendu par l’API est une liste de bières formatée comme suit :

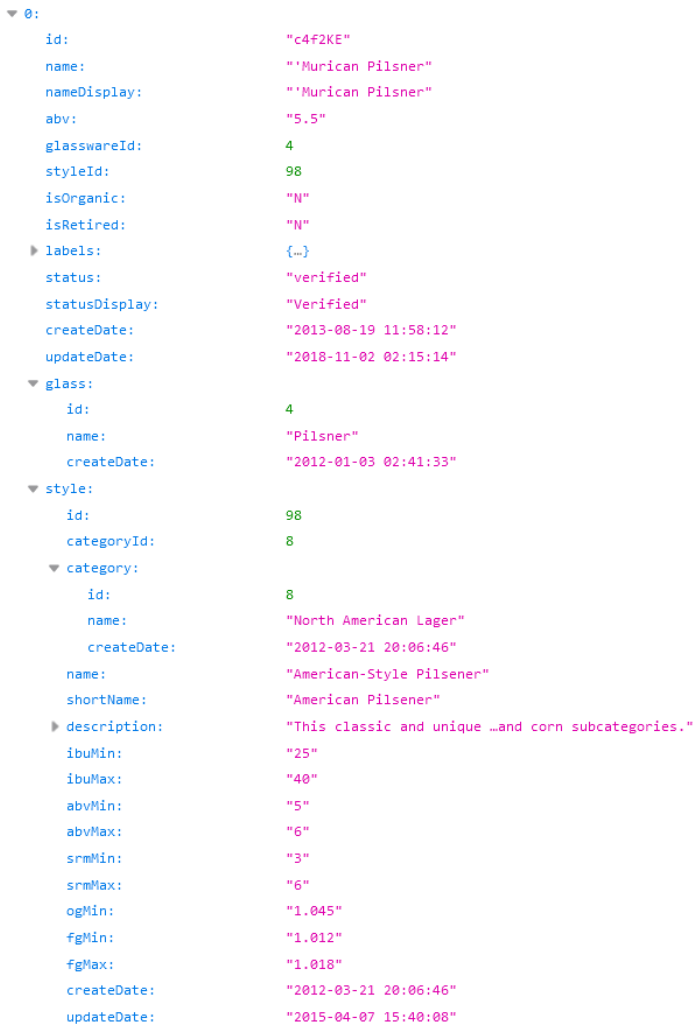

Figure 7: Résultat de l'API pour une bière

Les champs que nous utilisons sont « name » pour le nom, « abv » pour le taux d’alcool, « category>name » pour la catégorie, « status » pour le statut, « updateDate » pour la date de mise à

jour et « description » pour la description. Nous utilisons également l'image qui est un champ dans « labels »

5. Explications techniques

Au niveau de l'architecture du projet, nous avons fait le choix d'une architecture MVVM. Le ViewModel contient un Repository qui est connecté à la fois à Room et à l'API, ce qui nous permet d'obtenir des informations des deux sources uniquement en passant par le View Model. L'application contient deux activités. L'activité principale contient deux fragments : un pour l'affichage de la liste de toutes les bières et un pour l'affichage de le liste des bières favorites. L'autre activité correspond à l'affichage des détails d'une bière lors du clic sur celle-ci. Une Factory permet d'initialiser le ViewModel. La librairie Picasso est utilisée pour l'affichage des images et la barre de navigation est issue du GitHub d'Ismael Di Vita (https://github.com/ismaeldivita/chip-navigation-bar). Room est utilisé pour gérer la base de données locales et Retrofit est utilisé pour les appels asynchrones à l'API.

Le SDK minimum est le 24 et le SDK cible est le 30.