HMIN205
Développement logiciel pour mobiles

—

Réservation Service

—

Melvin Bardin
Malika Lin-Wee-Kuan

—

2020-2021

UNIVERSITÉ
DE MONTPELLIER

FACULTÉ DES SCIENCES
MONTPELLIER

Table des matières

Table des figures

1 Présentation du sujet

Dans le cadre du module HMIN205, notre objectif était de développer une application mobile permettant de réserver un service en ligne.

2 Liste des fonctionnalités

Nous allons vous présenter les fonctionnalités disponibles par type d'utilisateur.

2.1 Utilisateur invité (non connecté)

L'utilisateur non connecté à la possibilité de s'inscrire, de rechercher un service et de consulter les détails d'un service.

2.2 Client

L'utilisateur connecté en tant que client à la possibilité de se déconnecter, de rechercher des services ainsi que consulter leurs détails. Il a la possibilité de réserver un rendez-vous et de l'annuler. Il a la possibilité de consulter ses informations personnelles, les modifier et consulter les rendez-vous à venir.

2.3 Fournisseur

Le fournisseur possède les mêmes fonctionnalités que le client. Il peut en plus : créer, éditer, supprimer un service et consulter les clients qui ont réservé son service. Il a la possibilité d'annuler un rendez-vous.

3 Architecture de l'application

FIGURE 1 – Architecture de l'Application

L'application est centrée autour du tiroir de navigation, celle ci répertorie les activités accessibles par le type d'utilisateur utilisé. Sur la figure 1, on remarque bien que toutes les activités possèdent un tiroir de navigation sauf la page de connexion. La couleur des flèches indiquent le type d'utilisateur autorisé à accéder à l'activité *(rouge : uniquement les fournisseurs, vert : uniquement les utilisateurs connectés, noir : tous les utilisateurs)*. Le sens des flèches indiquent la transition possible d'une activité à une autre.

4 Les principales technologies / APIs utilisés

4.1 Room

Nous avons utilisé *Room* pour la base locale de l'application. Nous avons choisi cette technologie pour nous familiariser avec un outil important *d'Android*, plutôt que de nous disperser dans des technologies qui sortent du cadre du module. Toutes les activités sont munies d'une *ViewModel*, qui, à l'aide des *LiveData* actualisent l'interface lorsque les requêtes en base ont été effectuées.

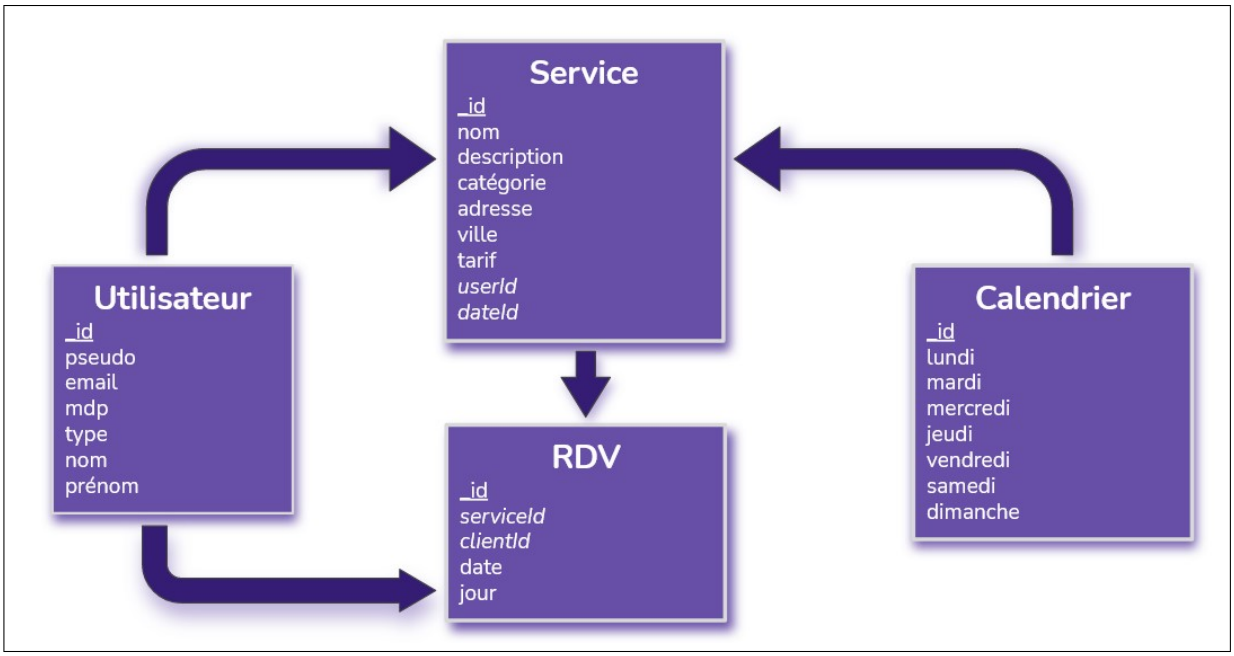

FIGURE 2 – Architecture de la base de données

4.2 Material Design

L'outil *Material Design* nous a permis d'incorporer des éléments complexes "facilement", comme notre tiroir de navigations.

4.3 CursorAdapter

Un *cursorAdapter* permet d'insérer les données d'un *Cursor* dans une *ListView*. Un *Cursor* est le type retourné par une requête *Room*. Lorsque celle ci retourne plusieurs éléments, un *Cursor* est donc comme un tableau qui contient le résultat de la requête.

4.4 Structuration optimale du code

Nous avons un maximum évité toute duplication de code dans notre projet, cela permet d'alléger les classes et le *XML*, mais cela facilite aussi la maintenance de l'application.

4.4.1 Des activités

Nous avons crée une classe abstraite pour toutes les activités avec un tiroir de navigation. Cette super-classe contient toutes les fonctions nécessaires au bon fonctionnement du tiroir de navigation.

4.4.2 Des ViewModels

Nos *ViewModels* héritent aussi d'une *super classe* qui permet d'initialiser la base de données et certains *LiveData*.

4.4.3 Du XML

Il est important de rappeler qu'un même élément dupliqué sur plusieurs activités doit posséder un identifiant différent, ce qui est très gênant. Nous avons donc isolé le code présent sur plusieurs activités dans des fichiers à part. Notamment pour la *topBar* et le tiroir de navigation qui sont présents sur toutes les activités.

4.5 Le Calendrier

Concernant le calendrier des disponibilités, nous sommes partis sur un choix de réservation par journée. Lors de la réservation d'un jour, on convertit la date du rendez-vous en numérique pour le mettre dans la base de données. Ainsi, lorsque nous recherchons les réservations à venir, il suffit de chercher les rendez-vous avec une date supérieure à celle actuelle.

5 Conclusion

5.1 Problèmes rencontrés et Améliorations possibles

Durant la réalisation de notre projet, nous avons rencontré quelques difficultés concernant la mise en place de la base de données ainsi que la vérification des requêtes effectuées. Nous avons eu aussi quelques soucis concernant le tiroir de navigation, car celui ci nécessitait beaucoup d'étapes à réaliser avant qu'il soit opérationnel.

En ce qui concerne les améliorations possibles, nous pourrions évidemment revoir notre calendrier, avoir un serveur pour stocker la base de données plutôt qu'une base de données locales, et aussi revoir le *design* de certaines activités.

5.2 Bilan

Pour ce projet, malgré le fait que nous n'avons pas un serveur et un *design* laissant à désirer sur certaines activités, nous sommes satisfaits de notre résultat. Surtout pour notre tiroir de navigation qui facilite énormément le déplacement entre les activités rendant l'expérience des utilisateurs agréable.