

Application Mobile : TicketApp

Sebastian Barrera Vega Juan, Chihi Zakaria,
Engelen Louis, Kamenga Lungeni Moïse et Stenhouse Quentin

2 mai 2017

Table des matières

| | | |
|---|-------------------------------------|---|
| 1 | Introduction | 2 |
| 2 | Travail d'équipe | 2 |
| 3 | Diagrammes | 2 |
| | 3.1 Diagrammes de Classes | 2 |
| 4 | Fonctionnement | 3 |
| | 4.1 Application | 3 |
| | 4.2 Base de donnée | 8 |
| 5 | Conclusion | 8 |

1 Introduction

Le but de notre application est de faciliter les transactions durant des événements. Un système largement adopté actuellement est le paiement par jetons ou tickets, plutôt que payer en liquide au bar. Ce système souffre d'un inconvénient majeur : d'un événement à l'autre, les tickets/jetons ne sont, en général, pas réutilisables. Ainsi, s'il y a une rupture de stock au cours de l'événement ou que l'événement finit avant que les clients aient utilisés tous leurs tickets/jetons, les clients s'en retrouvent lésés. TicketApp a pour but de palier à cet inconvénient en permettant un système de paiement facile.

Sur l'application, il est possible de voir le prix des boissons ou de la nourriture à l'événement. Le client peut ainsi décider avant la soirée combien de jetons/tickets il veut utiliser à l'événement. En payant en ligne, il peut mettre un crédit sur son compte. Tous ses paiements au cours de l'événement sont dès lors effectués via l'application.

Utiliser TicketApp palie à l'inconvénient cité plus haut. En effet, le solde restant sur le compte à la fin de l'événement est conservé. En conséquence, si un client revient à un autre événement organisé par le même groupe, collectif..., il peut réutiliser son crédit restant.

En plus de cet avantage pour l'utilisateur, c'est un avantage pour les organisateurs. De fait, cela leur permet non seulement de fidéliser les clients, mais aussi de prévoir leur stock en fonction des crédits prépayés. Cela représente un avantage certain dans la gestion des approvisionnements.

2 Travail d'équipe

Nous nous sommes organisés de la manière suivante :

- Dans un premier temps, nous avons déterminé les différents éléments qui étaient essentiels pour bien débiter le développement de l'application, par exemple : les diagrammes, la documentation nécessaire, etc. Après cela, nous nous sommes réparties les différents blocs et fixé des deadlines pour la soumission du travail produit sur *Github*.
- Pour la finalisation de l'application, tout un chacun d'entre nous a apporté sa pierre à l'édifice. En effet, chacun a résolu/apporté des solutions aux problèmes qui se posent pendant la finalisation d'un projet.

3 Diagrammes

3.1 Diagrammes de Classes

Dans le diagramme de classes qui suit, vous trouverez les classes qui représentent les métiers de l'application.

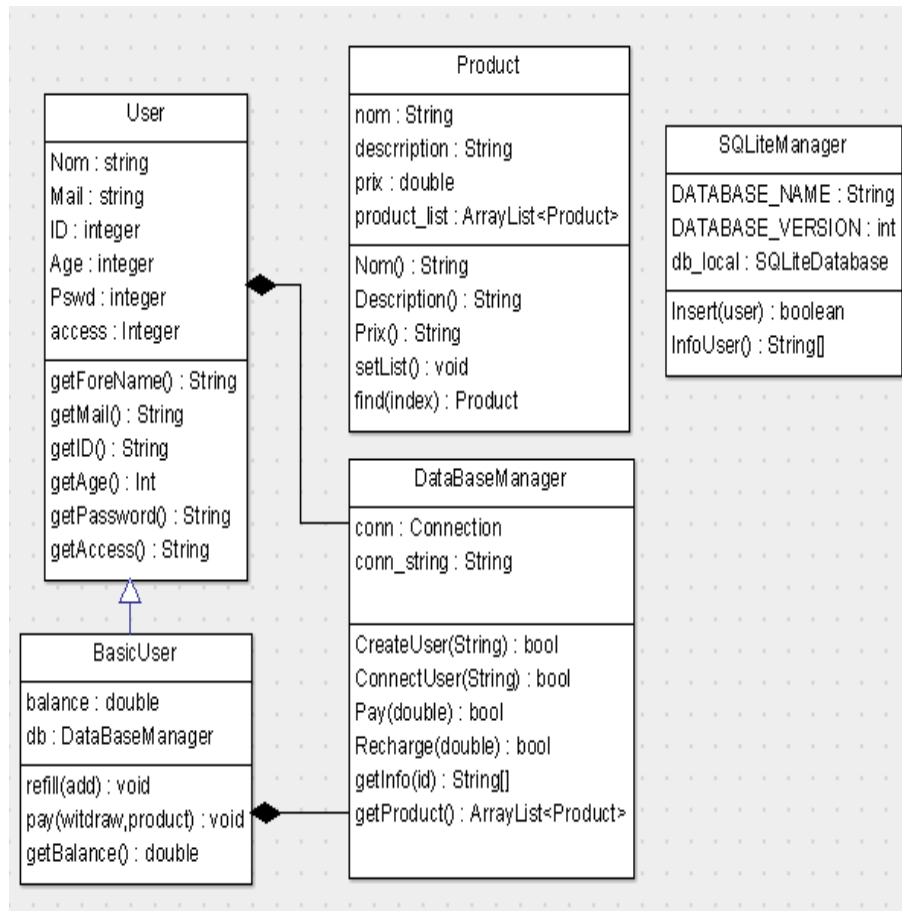


FIGURE 1 – Diagramme de classes

Comme vous pouvez e voir, nous avons les classes suivant :

- User/BasicUser : représente un utilisateur.
- Product : représente les produit disponible.
- DataBaseManager : gère les opération sur la base de données en ligne.
- SQLiteManager : gère les opération de la base de données locale.

4 Fonctionnement

4.1 Application

MainActivity

C'est le premier écran, l'activité d'accueil. Elle permet de se connecter ou d'avoir accès à la création d'un compte.

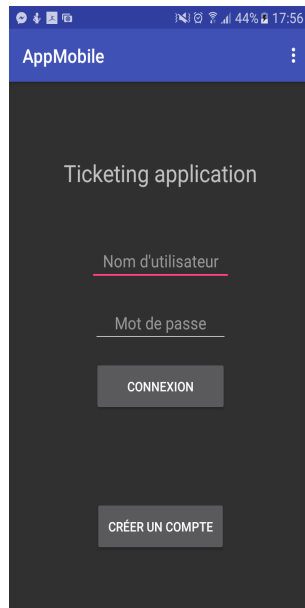


FIGURE 2 – MainActivity

UserActivity

Après connexion, cet écran s’affiche. Il reprend le nom, l’ID et le solde sur le compte. Elle permet aussi d’avoir accès aux écrans préférences, de recharge du solde et de listing des produits disponibles.

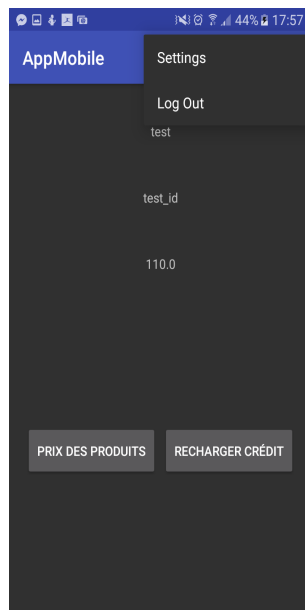


FIGURE 3 – UserActivity

AddUserActivity

Cet écran permet l'ajout d'un nouvel utilisateur. On y accède à partir de la MainActivity.

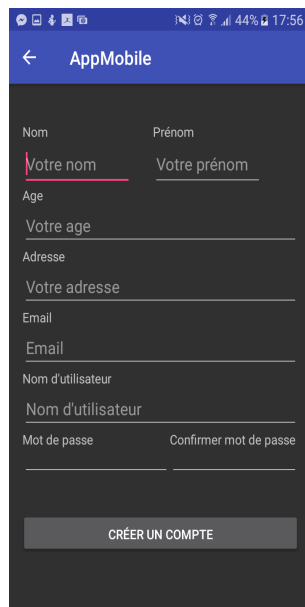


FIGURE 4 – AddUserActivity

ProductActivity

Cette activité contient le listing de tous les produits disponibles et permet d'accéder à l'écran de détail du produit sélectionné.

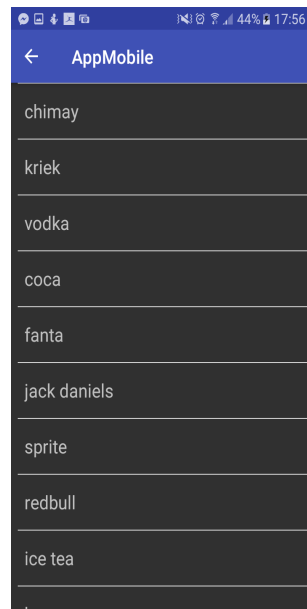


FIGURE 5 – ProductActivity

ProductDetails

Cet écran est accessible via la ProductActivity. Il permet de voir le nom, prix et autres détails de chaque produit. Il permet aussi de confirmer l'achat d'un produit.

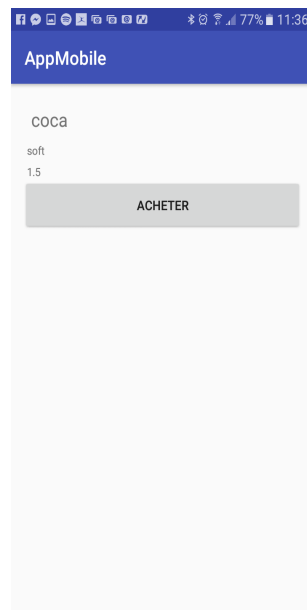


FIGURE 6 – ProductDetails

RefillActivity

Cet écran permet de recharger le solde du compte. Il est accessible via la UserActivity.

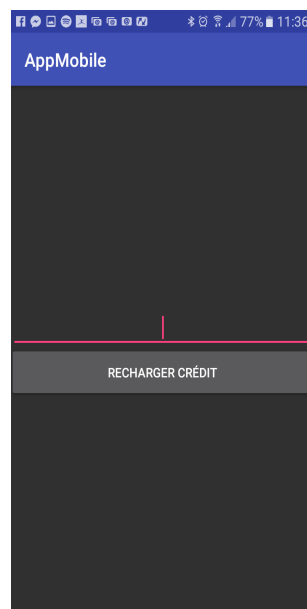


FIGURE 7 – Filtre LC de sortie

PreferenceActivity

Cette écran permet la modification de deux paramètres : la langue, la couleur.

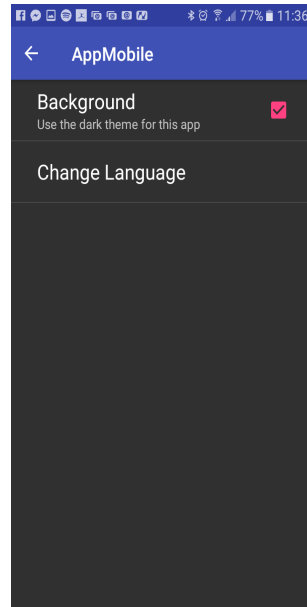


FIGURE 8 – Filtre LC de sortie

4.2 Base de donnée

LAMP

La communication avec la base de donnée représente la deuxième partie du projet. Cette dernière contient les différentes données nécessaires à l'application, par exemple, les informations des différents produits et utilisateurs.

SQLite

La SQLite assure la pérennité des informations nécessaires au bon fonctionnement de l'application sans devoir passer par la base de donnée en ligne.

5 Conclusion

La partie application est complètement fonctionnelle, toutes les fonctionnalités ont été développées. Certains aspects de l'application devront toutefois être sujet à modification, correction ou devront être améliorées. Ceci touche principalement la communication avec la base de donnée en ligne.