

Projet Android Fil Rouge

Alexander BORETTI

Rayane LARBI

Jérémy MONCADA

URL Vidéo : <https://youtu.be/KzCaVO-Udb8>

Boredeals est une application payante (7,99 €) téléchargeable sur le PlayStore où l'on vend des services. Cette application permet d'avoir accès à différents code promos allant de -10% jusqu'à -50% sur un large catalogue de différentes marques.

Notre application se divise en 7 activités. Tout d'abord, il y a une activité permettant de se connecter et de s'inscrire. Ensuite, nous avons l'activité principale, qui est la page d'accueil où se trouve le catalogue regroupant les différentes marques.

La deuxième activité principale est celle des "codes promo". À l'intérieur, on trouve le code promotionnel de la marque en question ainsi que la réduction en pourcentage associée.

Enfin, la dernière activité est celle des "favoris". Dans cette section, vous trouverez la liste des marques que vous avez ajoutées à vos favoris.

Nous disposons également d'une activité permettant de gérer les QR codes et les magasins de marque.

Page de login

C'est la première page que vous verrez lorsque vous lancerez l'application. Vous pouvez soit vous connecter si vous avez déjà un compte (les données sont enregistrées dans Firebase), soit vous inscrire en saisissant votre nom, votre adresse électronique, votre pseudonyme et votre mot de passe.

Page d'accueil

Un ImageView représentant un cœur animé se trouve en haut à gauche de cette activité, permettant de retrouver toutes les marques que l'on aura ajoutées à nos favoris.

Une RatingBar est positionnée sous le titre de notre application pour filtrer les codes promo de la plus petite réduction à la plus grande.

Dans les différents widgets de chaque marque, on trouve un bouton "J'en profite" cliquable, qui nous redirige vers la page du code promo de la marque en question.

Un ImageButton, représenté par un cœur, nous permet d'ajouter la marque en question à nos favoris.

Chaque widget indique le nom de la marque et son code promo de réduction.

Page des magasins

Dans cette activité, un widget est créé pour chaque magasin, comprenant un titre, une description, un bouton "J'en profite" et enfin un cœur permettant d'ajouter les marques aux favoris.

Un RecyclerView, situé dans l'activité principale, génère ensuite plusieurs widgets à partir de cette activité. Enfin, dans le fichier MainActivity.java, nous avons une liste qui récupère les données situées dans le fichier strings.xml (le titre, la description et le pourcentage de l'offre) afin d'afficher et de lier les données correspondantes à chaque marque.

Page des codes promos

Cette activité affiche le nom de la marque concernée, le pourcentage de l'offre, une RatingBar permettant d'évaluer le code promo, le code promo lui-même avec un bouton permettant de le copier dans le presse-papiers, et enfin, un bouton permettant de générer un QR code pour accéder au site de la marque.

Page des favoris

Cette dernière activité affiche simplement la liste des marques que l'on aura mises en favoris.

Cahier des charges

Animation

Pour cette application, nous avons utilisé une animation de vue pour le cœur animé en haut à gauche de la page d'accueil, ainsi qu'un ObjectAnimator dans l'activité des codes promo, notamment pour agrandir le code promo une fois arrivé sur la page.

Concernant le bouton "Copier et utiliser le code promo", nous avons mis en place un feedback pour indiquer clairement que le code a été copié avec succès, à l'aide d'un Toast.

Nous avons également ajouté une animation à la RatingBar dans l'activité des codes promo, pour fournir un feedback lorsque l'utilisateur clique sur l'un des coeurs.

Nous avons choisi d'utiliser ces deux types d'animations car ils semblaient les plus cohérents pour notre équipe. Cela nous a permis de tester et de découvrir les animations d'ObjectAnimator.

SeekBar / RatingBar

Pour la SeekBar, nous avons utilisé celle présentée et étudiée lors du TP. Elle filtre simplement les codes promo du plus petit au plus grand.

En ce qui concerne la RatingBar, nous avons décidé de la personnaliser. Nous utilisons deux fichiers XML (heart fill et empty) dans le dossier drawable, combinés dans un autre fichier XML (combi) avec des layer-list. Cela nous permet d'avoir une progression personnalisée de la RatingBar.

Nous avons opté pour cette approche car notre application ne nécessite pas plus de SeekBars ni de RatingBars.

Plusieurs Activités

Nous avons développé un total de 7 activités, justifiant ce choix en fonction de notre application.

Étant donné que nous proposons des services payants, nécessitant un accès aux codes promos moyennant un paiement pour l'application, nous n'avons pas jugé nécessaire de créer davantage d'activités. Une seule vue d'accueil était également suffisante pour répondre à nos besoins.

Echange de données

Un Singleton offre un accès global à une seule instance de la classe, permettant ainsi le partage des données entre toutes les parties de l'application. Il est approprié pour stocker des données qui doivent être accessibles de manière globale et partagées entre différentes parties de l'application.

Contrairement au Singleton, Parcelable est utilisé pour transmettre des objets complexes entre les composants Android, tels que les activités. Il est adapté pour passer des données spécifiques à travers les frontières des activités.

Nous avons donc choisi d'utiliser Parcelable dans ce cas, car il offre une solution plus standard et propre pour transmettre des données spécifiques entre les activités. Cela rend également le code plus compréhensible pour les autres développeurs travaillant sur le projet et suit les bonnes pratiques Android.

Json et webService

Pour cette partie de l'application, nous avons utilisé la bibliothèque d'objets Java au format JSON.

Elle nous a permis de manipuler facilement des données dans ce format, offrant des fonctionnalités de sérialisation et de désérialisation qui simplifient l'échange, le stockage et le traitement des données par rapport à une approche plus traditionnelle.

Cette bibliothèque s'est avérée très utile pour l'échange de données.

En ce qui concerne le chargement et l'affichage des images, nous avons utilisé la bibliothèque Picasso, qui est open-source. Elle simplifie le processus de téléchargement et de mise en cache d'images. Nous récupérons nos ressources à partir des URL spécifiées.

Technologie non étudiée

Le QR code est une technologie de plus en plus répandue mais souvent sous-estimée dans notre vie quotidienne. Bien que de nombreuses personnes connaissent son existence, peu comprennent pleinement son potentiel et les nombreuses possibilités qu'il offre.

Le QR code est une technologie non seulement sous-estimée, mais aussi mal comprise. Son potentiel reste largement inexploité dans de nombreux domaines. Pourtant, à mesure que les consommateurs deviennent de plus en plus familiers avec cette technologie, son adoption et son utilisation créative continueront de croître.

Son utilisation nous a permis de comprendre le fonctionnement des QR Codes.

Répartition des tâches

Tâches	Alexander.B	Rayane.L	Jérémy.M
Codage de l'activité main.xml			
Codage de l'activité code.xml			
Codage de l'activité store.xml			
Codage de l'activité QRCode.xml			
Codage de l'activité login.xml			
Codage de l'activité signUp.xml			
Codage de l'activité fav.xml			
RatingBar Personnalisée des codes promos			
Animation de battements de cœur			
Animation ObjectAnimator CodePromo			
SeekBar de la page d'accueil			
Classe parcelable			
Json et webService			
Firebase			
NB Total Heures	20H	20H	20H

60 H