**Documentation du Projet Android Kotlin - Application de Produits avec Panier et QR Code**

## 🧾 Sommaire

1. [Présentation générale](#pr%C3%A9sentation-g%C3%A9n%C3%A9rale)
2. [Architecture de l’application](#architecture-de-lapplication)
3. [Fonctionnalités principales](#fonctionnalit%C3%A9s-principales)
4. [Structure des packages](#structure-des-packages)
5. [Détails des écrans](#d%C3%A9tails-des-%C3%A9crans)
6. [VueModèle (ViewModel)](#vuemod%C3%A8le-viewmodel)
7. [QR Code – Génération & Utilisation](#qr-code--g%C3%A9n%C3%A9ration--utilisat)
8. [Navigation](#navigation)
9. [Améliorations possibles](#am%C3%A9liorations-possibles)
10. [Dépendances importantes](#d%C3%A9pendances-importantes)

## 📌 Présentation générale

L'application permet :

* d’afficher une liste de produits,
* de consulter les détails de chaque produit,
* d’ajouter des produits au panier,
* de modifier les quantités,
* de générer un QR code menant à la page de détail d’un produit.

Technologies utilisées :

* Kotlin
* Jetpack Compose (Material 3)
* Navigation Compose
* Coil pour le chargement d’images
* ZXing pour les QR codes

## 🏛️ Architecture de l’application

**Architecture MVVM** :

* Model → objets Product, CartItem
* ViewModel → logique de gestion des produits et du panier
* View → Composables (ProductScreen, CartScreen, etc.)

## ✨ Fonctionnalités principales

| **Fonction** | **Description** |
| --- | --- |
| 📋 Liste de produits | Affichage des produits avec filtre et recherche |
| 🔍 Détail produit | Image, prix, description, bouton "Ajouter au panier" |
| 🛒 Panier | Ajout/suppression/quantité, total dynamique |
| 📷 QR Code | Généré pour chaque produit pour le scanner ultérieurement |
| 🧭 Navigation | Entre écran d’accueil, panier, détail produit |

## 📁 Structure des packages

kotlin

CopierModifier

com.example.projet\_androidkotlin

│

├── data/

│ └── model/ → Product.kt, CartItem.kt

│

├── ui/

│ └── screens/ → ProductScreen.kt, CartScreen.kt, ProductDetailScreen.kt

│

├── utils/ → generateQRCode.kt

│

├── viewmodel/ → ProductViewModel.kt

## 🖼️ Détails des écrans

**ProductScreen.kt**

* Affiche tous les produits dans une LazyColumn
* Barre de recherche
* Menu de filtrage : prix croissant, décroissant, ordre alphabétique
* Boutons : accueil, panier

**ProductDetailScreen.kt**

* Détail du produit (image, titre, prix, description)
* Bouton “Ajouter au panier”
* Boutons d’accès rapide à l’accueil et au panier
* Intégration du QR code possible via ProductQRCode()

**CartScreen.kt**

* Liste des produits dans le panier
* Boutons :
  + + : augmente la quantité
  + - : diminue la quantité
  + Supprimer : retire complètement
* Total affiché automatiquement
* Navigation : retour & accueil

## 🧠 VueModèle (ProductViewModel.kt)

Contient :

* products: liste complète chargée (flow)
* cartItems: état du panier (flow)
* product: produit sélectionné pour détail

Fonctions principales :

kotlin

CopierModifier

fun loadProducts()

fun loadProductById(id: Int)

fun addProductToCart(product: Product)

fun increaseQuantity(product: Product)

fun removeFromCart(product: Product)

fun removeProductCompletely(product: Product)

## 🔳 QR Code – Génération & Utilisation

**Fichier : generateQRCode.kt**  
Fonction pour créer un Bitmap à partir d'une chaîne (URL, ID produit, etc.)

kotlin

CopierModifier

fun generateQRCode(content: String): Bitmap

**Utilisation dans un Composable :**

kotlin

CopierModifier

@Composable

fun ProductQRCode(productUrl: String) {

val qrBitmap = remember(productUrl) { generateQRCode(productUrl) }

Image(bitmap = qrBitmap.asImageBitmap(), contentDescription = null)

}

👉 Tu peux appeler cette composable dans ProductDetailScreen.

## 🧭 Navigation

Définie avec NavHost :

kotlin

CopierModifier

NavHost(navController, startDestination = "products") {

composable("products") { ProductScreen(...) }

composable("cart") { CartScreen(...) }

composable("productDetail/{productId}") { backStackEntry ->

val productId = backStackEntry.arguments?.getString("productId")

ProductDetailScreen(productId = productId, ...)

}

}

## 📦 Dépendances importantes

kotlin

CopierModifier

// Compose

implementation("androidx.compose.material3:material3:1.2.0")

implementation("androidx.navigation:navigation-compose:2.6.0")

// Coil pour les images

implementation("io.coil-kt:coil-compose:2.3.0")

// ZXing pour QR code

implementation("com.google.zxing:core:3.5.2")

// Lifecycle & ViewModel

implementation("androidx.lifecycle:lifecycle-viewmodel-compose:2.6.2")

implementation("androidx.lifecycle:lifecycle-runtime-compose:2.6.2")