

**Projet Android Kotlin - Documentation**

# **Réalisé par**

**Jules Paolantoni**

**Julien Thepaut**

# Android Developers Blog: Android Studio @ I/O '23: Announcing Studio Bot, an AI-powered coding assistant **Sur**

# **🧾Sommaire**

[**Présentation générale 2**](#_Toc199588467)

[**Architecture de l’application 2**](#_Toc199588468)

[**Fonctionnalités principales 3**](#_Toc199588469)

[**Structure du projet 3**](#_Toc199588470)

[**Détails des écrans 4**](#_Toc199588471)

[**VueModèle (ProductViewModel.kt) 5**](#_Toc199588472)

[**QR Code – Génération & Utilisation 5**](#_Toc199588473)

[**Navigation 6**](#_Toc199588474)

[**Dépendances importantes 6**](#_Toc199588475)

# **Présentation générale**

**L'application permet de :**

* Afficher une liste de produits,
* Consulter les détails de chaque produit,
* Ajouter des produits au panier,
* Modifier les quantités,
* 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 du projet**

com.example.projet\_androidkotlin

├── data/ → Contient les classes liées aux données de l'application

│ └── model/ → Définit les modèles de données utilisés dans l'app

│ ├── Product.kt → Représente un produit avec ses attributs (titre, prix, image, etc.)

│ └── CartItem.kt → Représente un article dans le panier (produit + quantité)

├── ui/ → Contient tout ce qui est lié à l'affichage (interface utilisateur)

│ └── screens/ → Écrans de l'application affichés à l'utilisateur

│ ├── ProductScreen.kt → Affiche la liste des produits disponibles

│ ├── CartScreen.kt → Affiche les articles ajoutés au panier

│ └── ProductDetailScreen.kt → Affiche les détails d’un produit sélectionné

├── utils/ → Fonctions utilitaires utilisées dans différentes parties du projet

│ └── generateQRCode.kt → Génère un QR Code à partir d’un texte

│ (ex: lien ou référence produit)

├── viewmodel/ → Contient les ViewModels pour gérer les données de l'UI

│ └── ProductViewModel.kt → Gère les états et les données liés aux produits et au

panier

# **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)}

# **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")