

DOCUMENTATION TECHNIQUE ET FONCTIONNELLE D'UNE APPLICATION ANDROID DE GESTIONNAIRE DE COLLECTION DE CD & VINYLES.

Emmanuel LE MOUËLLIC

CDA 2023 - GRETA de Vannes

Document Technique - Application Record Store

Introduction

L'application "Record Store" est une application mobile destinée aux amateurs de musique. Elle permet aux utilisateurs de gérer leur collection de disques de manière numérique.

L'application offre plusieurs fonctionnalités, notamment la possibilité d'ajouter des disques à la collection en scannant leur code-barres, de visualiser les détails de chaque disque et de gérer l'ajout ou la suppression des médias de la collection dynamiquement.

Le but de l'application est de fournir une plateforme facile à utiliser pour les collectionneurs de disques, leur permettant de gérer et d'organiser leur collection de manière efficace.

Architecture de l'application

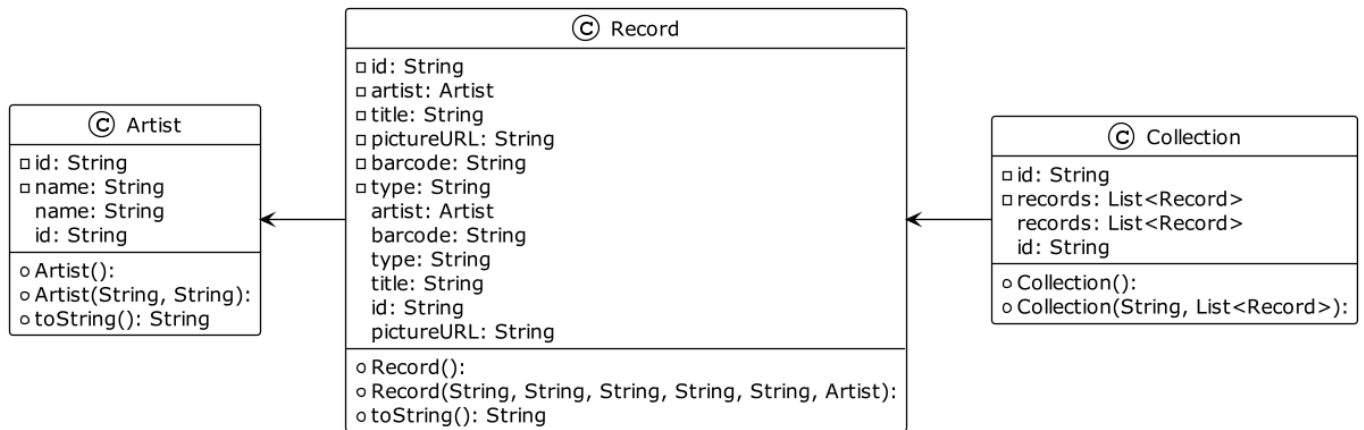
Objets métiers :

Classe Artist : Cette classe représente un artiste de musique. Chaque artiste est caractérisé par un identifiant unique et un nom. Ces informations sont récupérées depuis l'API MusicBrainz.

Classe Record : Cette classe représente un disque de musique. Chaque disque est caractérisé par un identifiant unique, un titre, une URL d'image pour la pochette, un format (album, vinyle, etc.), un code-barres et un artiste. L'artiste est représenté par une instance de la classe Artist, ce qui signifie que la classe Record encapsule la classe Artist. Cela signifie que chaque disque est associé à un et un seul artiste.

Classe Collection : Cette classe représente une collection de disques de musique. Chaque collection est caractérisée par un identifiant unique et une liste de disques. Les disques sont représentés par des instances de la classe Record, ce qui signifie que la classe Collection encapsule la classe Record. Chaque collection peut contenir plusieurs disques.

Voici un schéma UML illustrant ceci:



Use Case:

Se connecter : L'utilisateur peut se connecter à l'application. Le système fournit l'authentification nécessaire via l'API de Firebase.

Afficher la collection : Une fois connecté, l'utilisateur peut afficher sa collection de disques. Le système fournit les données nécessaires pour afficher la collection.

Utiliser l'application principale : Après avoir affiché sa collection, l'utilisateur peut utiliser les fonctionnalités principales de l'application, comme ajouter ou supprimer des disques de sa collection.

Voici un schéma détaillant le use case:

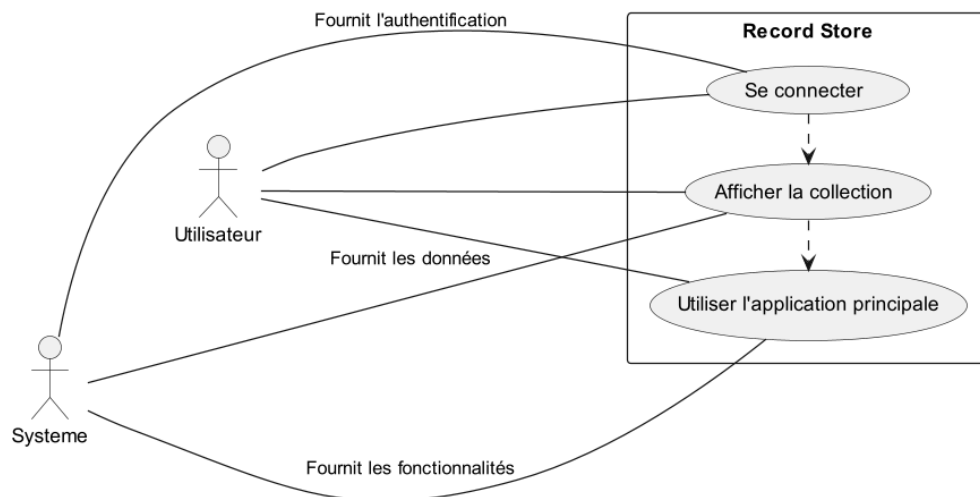
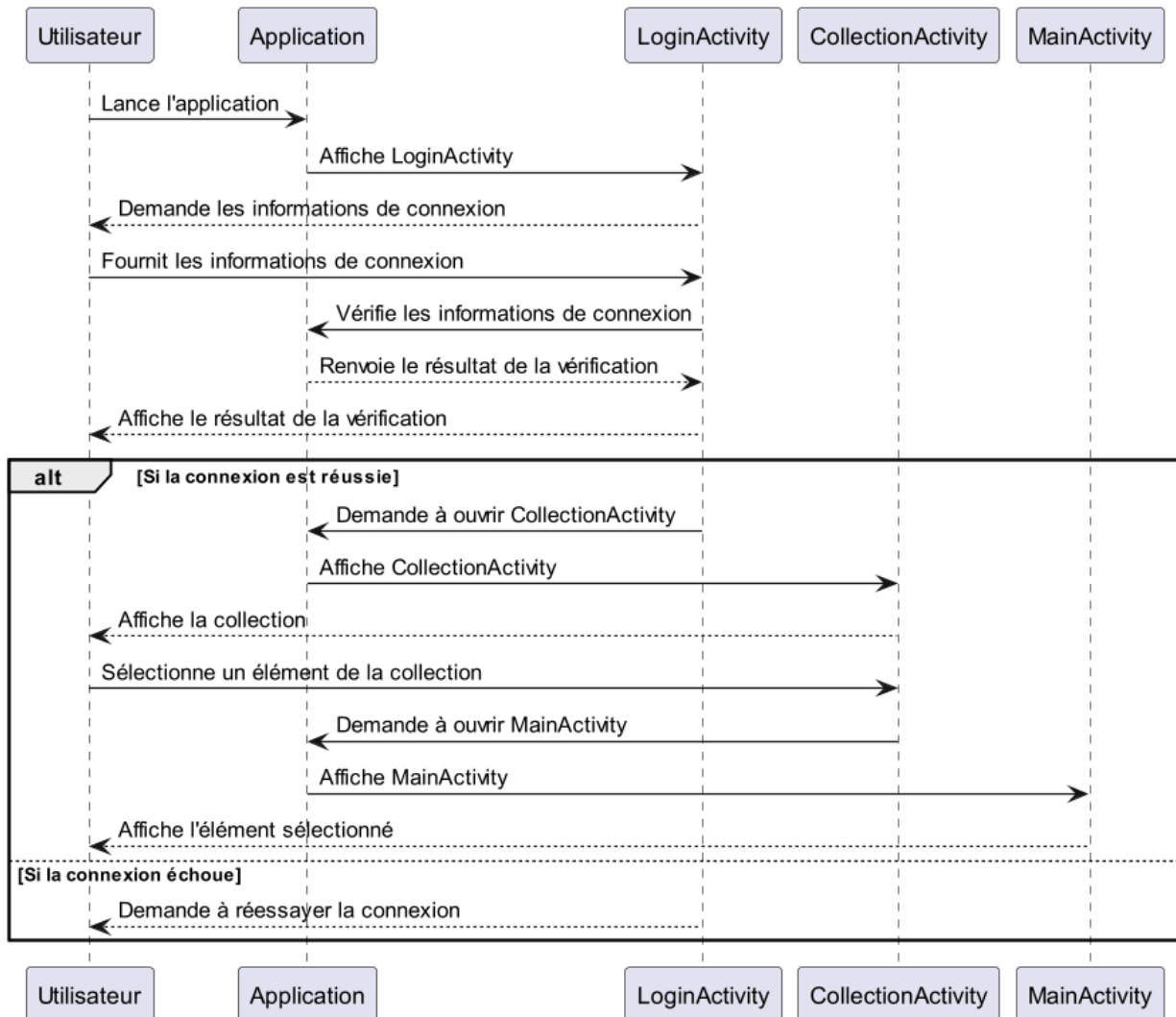


Diagramme de séquence:



Stack technique

Langage de programmation :

Java

Frameworks :

Android SDK basé sur Android 14 UpsideDownCake API Level 34 .

Outils de gestion de dépendances :

Gradle

Environnement de développement :

Android Studio Hedgehog | 2023.1.1

Base de données :

Firebase Realtime Database

Authentification :

Firebase Auth est utilisé pour l'authentification des utilisateurs.

API :

MusicBrainz

Autres outils :

Glide pour la mise en page des images.

ZXing ("Zebra Crossing") pour la lecture des codes-barres.

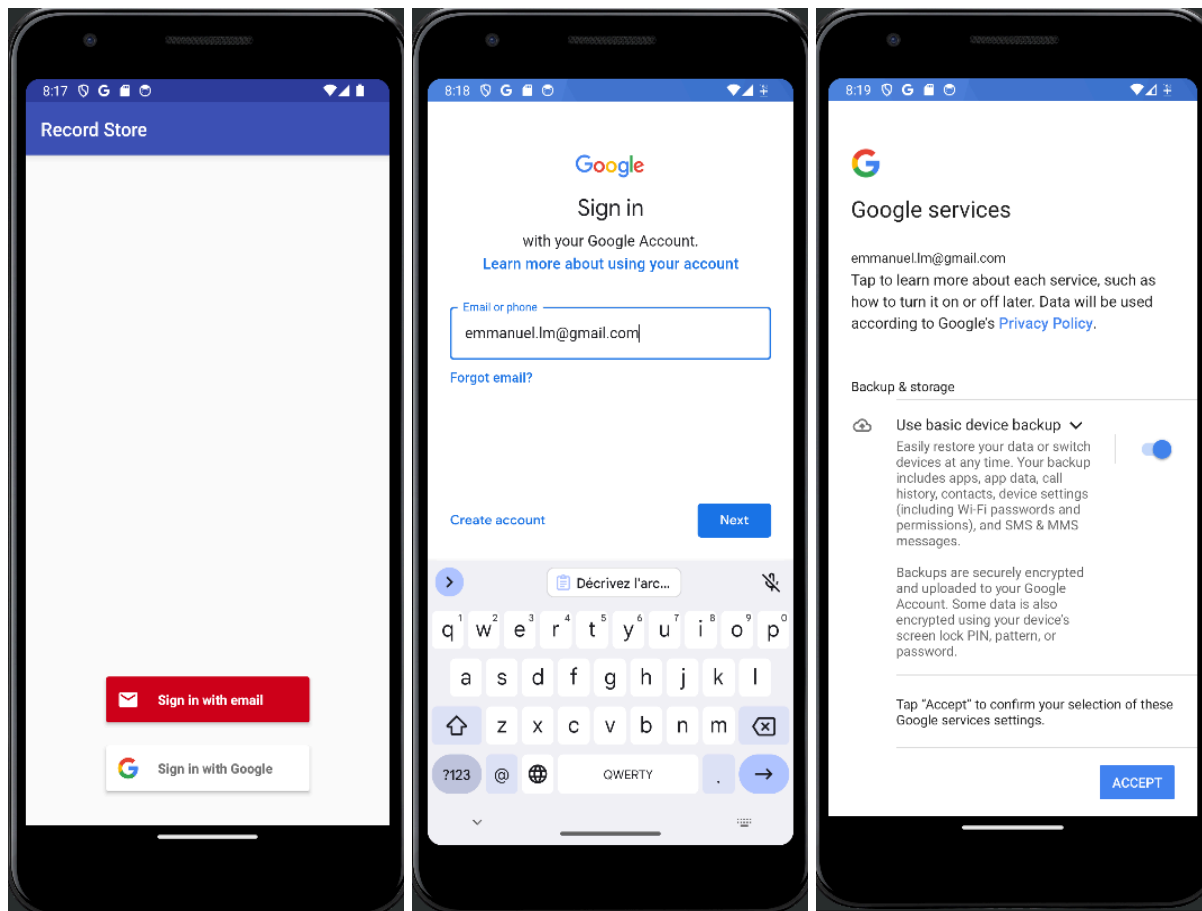
Gestion de version :

Git hébergé sur Github

Parcours utilisateur

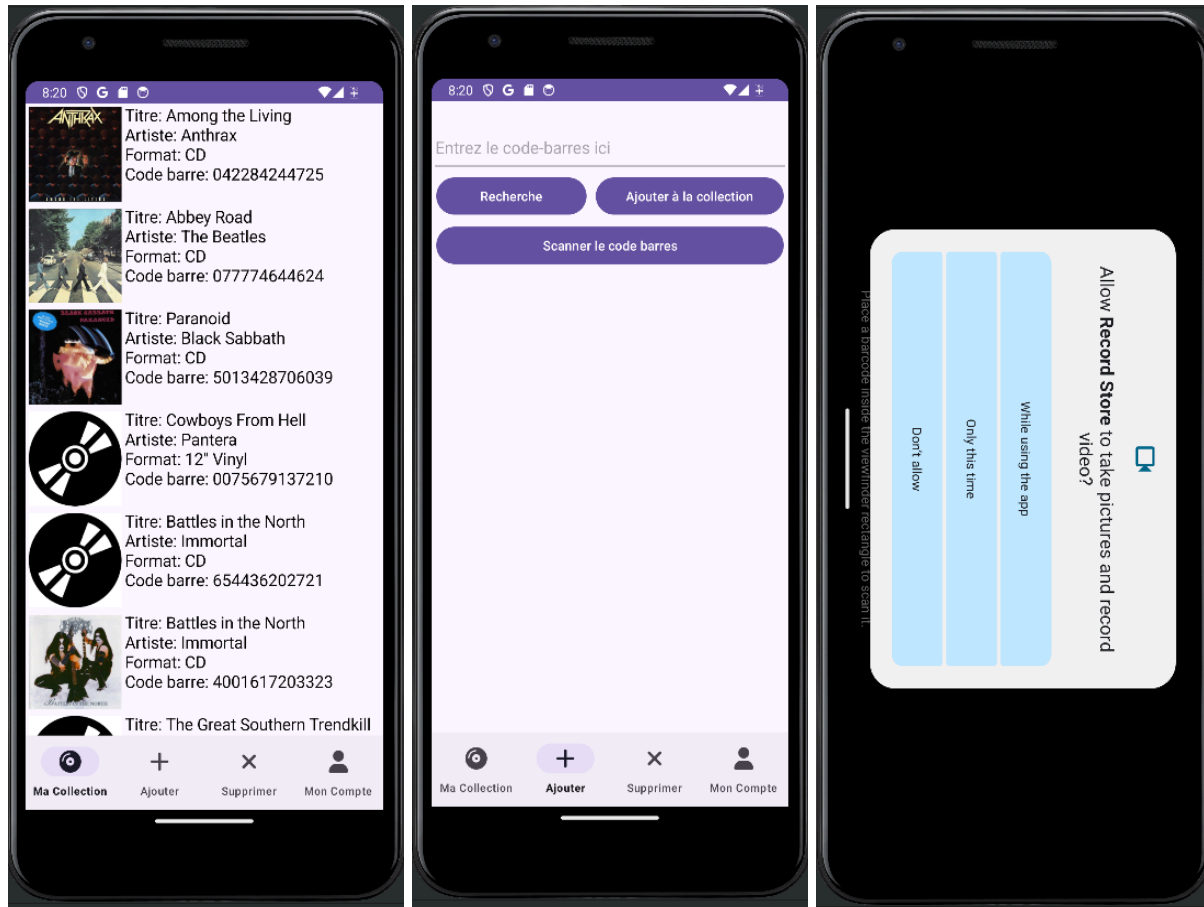
Authentication:

L'utilisateur est invité à se connecter via son compte Google



Collection:

L'utilisateur arrive sur sa collection, il peut ajouter un nouveau média en saisissant le code barres ou en le scannant via la bibliothèque ZXing, il doit alors accorder la permission à l'application d'utiliser la caméra du smartphone



Gestion de la collection et de l'utilisateur:

L'utilisateur peut alors gérer sa collection en supprimant les entrées existantes de la base de données. Une alerte est alors envoyée pour confirmer la suppression du média de la collection. Si il le souhaite, il peut se déconnecter de l'application via le menu Mon Compte.

