

Gabriel Huygebaert

Innovations technologiques

Projet : Site de vente d’objets d’occasion

FreeBay

Enseignant : G. Wetteren

# Énoncé

**[Innovations technologiques]**

*Projet : Site de vente d’objets d’occasion*

Il est demandé de développer une application Android en Java permettant d’acheter des objets en seconde main.

Un utilisateur inscrit et connecté pourra :

* ajouter un objet à vendre
* obtenir la liste des tous les objets et filtrer cette liste en n’affichant que les objets proches de lui
* visualiser les informations détaillées d’un objet
* placer un objet dans sa liste des objets à suivre et afficher sa liste d’objets à suivre

Les différentes informations seront stockées dans une base de données. Cette base de données sera manipulée par des services distants (RPC PHP ou services web RESTfull en Java) qui renverront des informations au format texte pur ou au format JSON.

[…]

Table des matières

[Énoncé 1](#_Toc111475903)

[Façon d’utiliser le programme 2](#_Toc111475904)

[Comptes déjà existants 2](#_Toc111475905)

[Account 2](#_Toc111475906)

[Lien Git 2](#_Toc111475907)

[Base de données 3](#_Toc111475908)

[Jet 1 3](#_Toc111475909)

[Analyse 3](#_Toc111475910)

[Use case diagram 3](#_Toc111475911)

[Jet 1 3](#_Toc111475912)

[Class diagram 3](#_Toc111475913)

[Jet 1 3](#_Toc111475914)

# Faire fonctionner le programme

## Création de la base de données

Depuis UwAmp, accéder à l’outil PHPMyAdmin et créer une nouvelle base de données vide.

Une image contenant texte

Description générée automatiquement

Création des tables => sgbd/CREATE.sql

## Modifications pour accéder aux RPC php

### Dans le programme

#### Package be.huygebaert.freebay.accessDb.Config.java

Les membres utiles à la constitution de l’objet URL ont été gérées dans cette classe mère.

Il vous faut copier/coller le dossier « android » présent dans .rar, qui contient les rpcs PHP de mon projet, dans le /UwAmp/www.

#### /android/Config.php

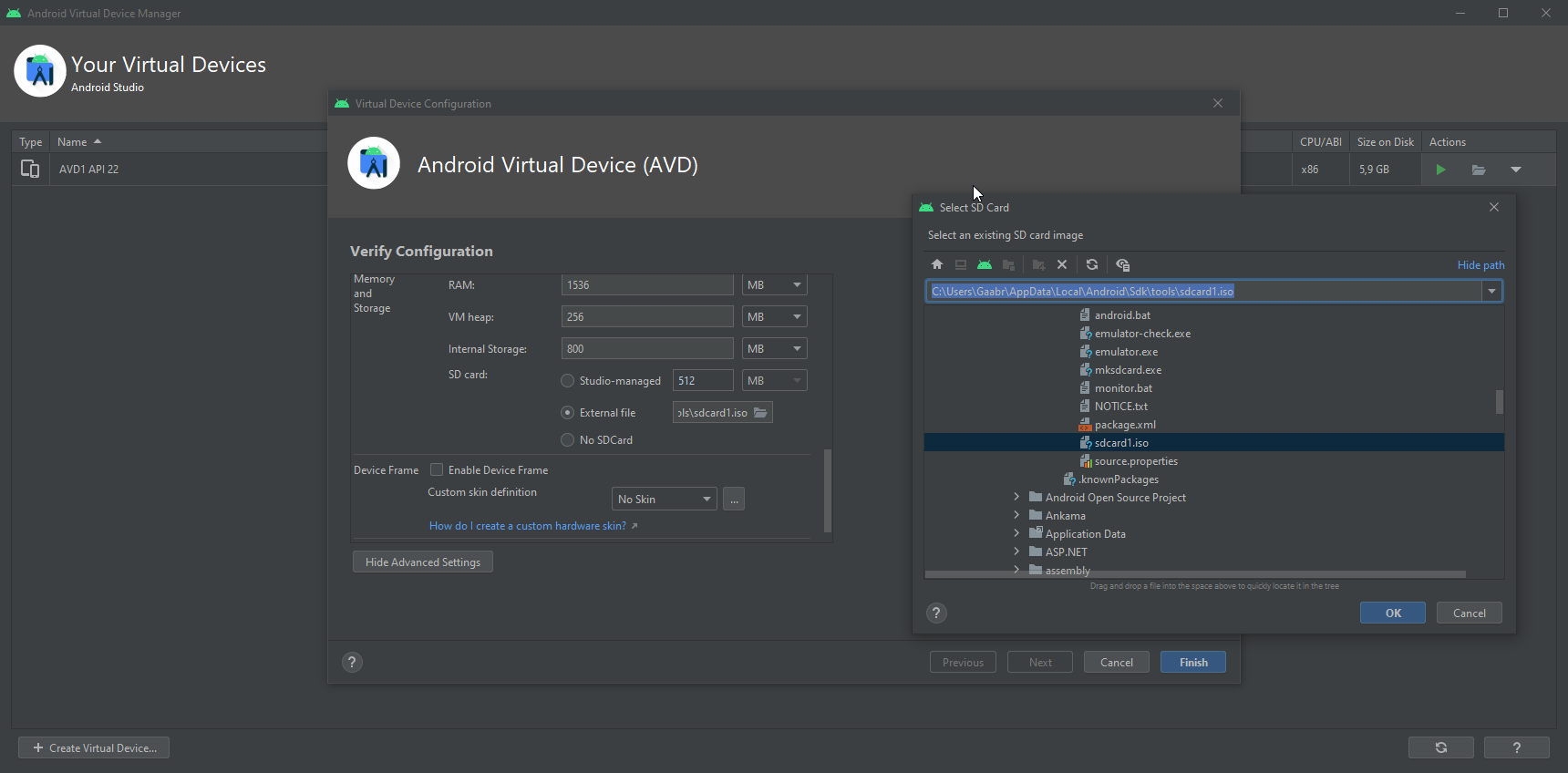
De la même manière, les variables utiles aux appels à la base de données par les RPC sont centralisées.

#### Emuler une carte SD (si nécessaire, pour ajouter des images )

Mon AVD ne possédait pas de carte SD. Possiblement, cela sera votre cas aussi.

J’ai monté la carte SE en fat 32 grâce à la commande : *C:\Users\Gaabr\AppData\Local\Android\Sdk\tools>mksdcard 1024M sdcard1.iso*

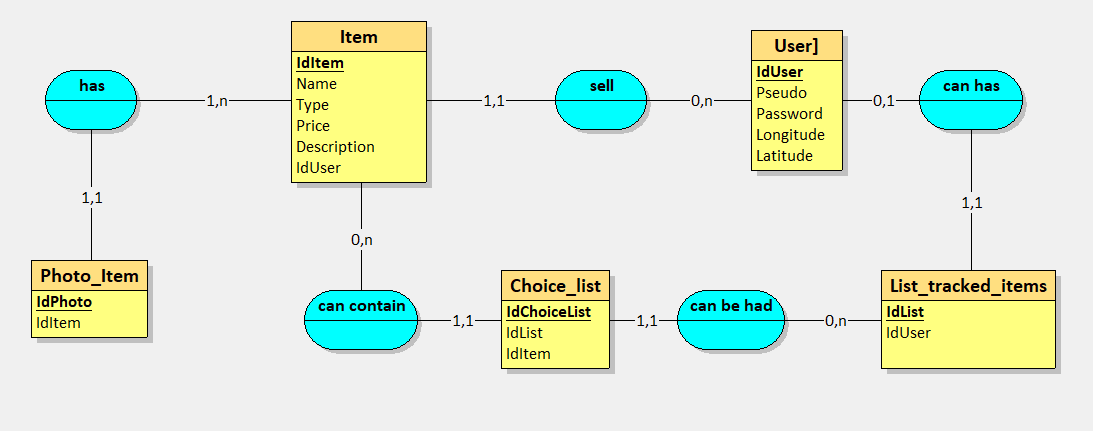
Depuis l’AVD manager dans android studio, édit l’AVD en question -> show advanced settings -> external File -> choisir le .iso créé juste avant.



# Lien Git

https://github.com/Kira-Atha/Android2-22/tree/Gabriel

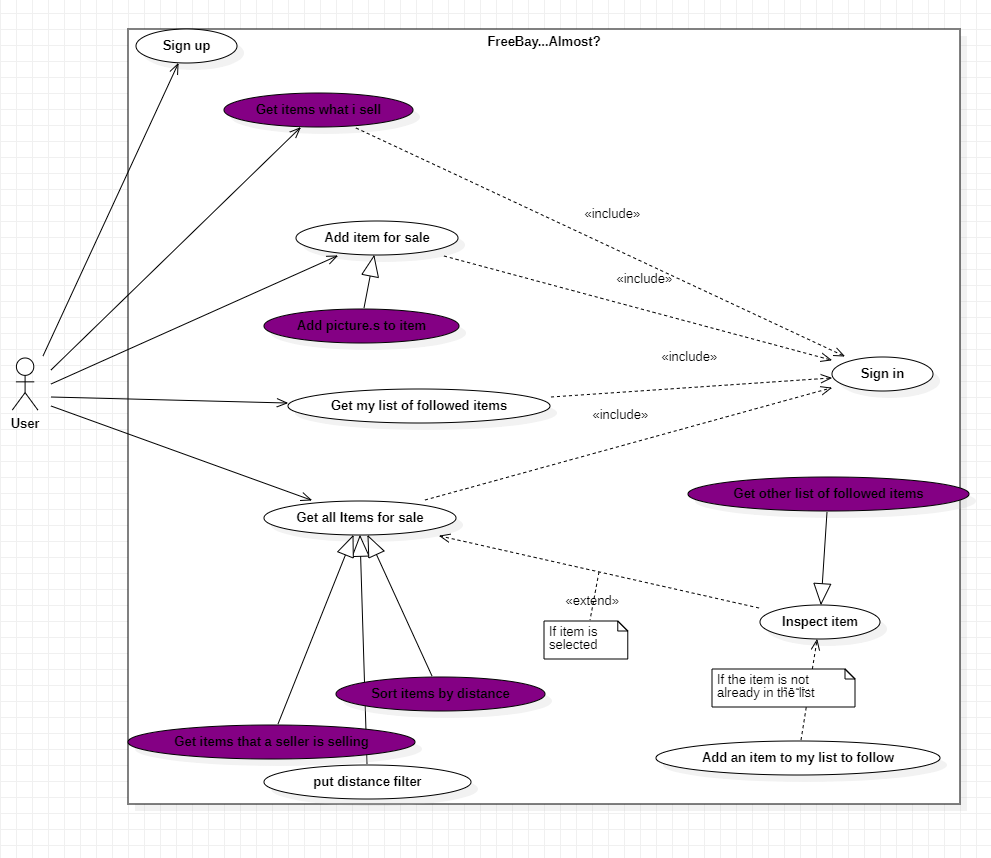
# Base de données



La table « list\_tracked\_items » permet d’anticiper une éventuelle amélioration du programme. Par exemple, donner la possibilité à l’utilisateur de créer plusieurs listes ( Alors, pourquoi pas un ajout du nom de la liste aussi ?).

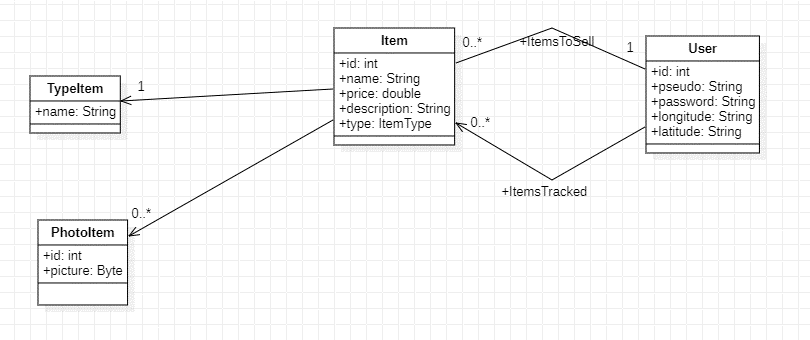
# Analyse

## Use case diagram



Afin de présenter la personnalisation du projet, voici le diagramme des cas d’utilisations. Les use cases violets sont ceux qui ont été ajouté par rapport à l’analyse de base.

## Class diagram



Un double lien de l’utilisateur à la classe objet permet de scinder les listes. L’utilisateur possède donc deux références d’items. La première concerne ceux qu’il vend. La deuxième concerne ceux qu’il suit. Pour un objet mis en vente, il est intéressant de connaître son vendeur afin de calculer la distance entre ce dernier et la personne qui consulte les détails de l’objet. En revanche, il n’est pas utile de savoir par qui il est suivi, puisque nous savons quels objets sont suivis par un utilisateur.