

Projet final - Base de données

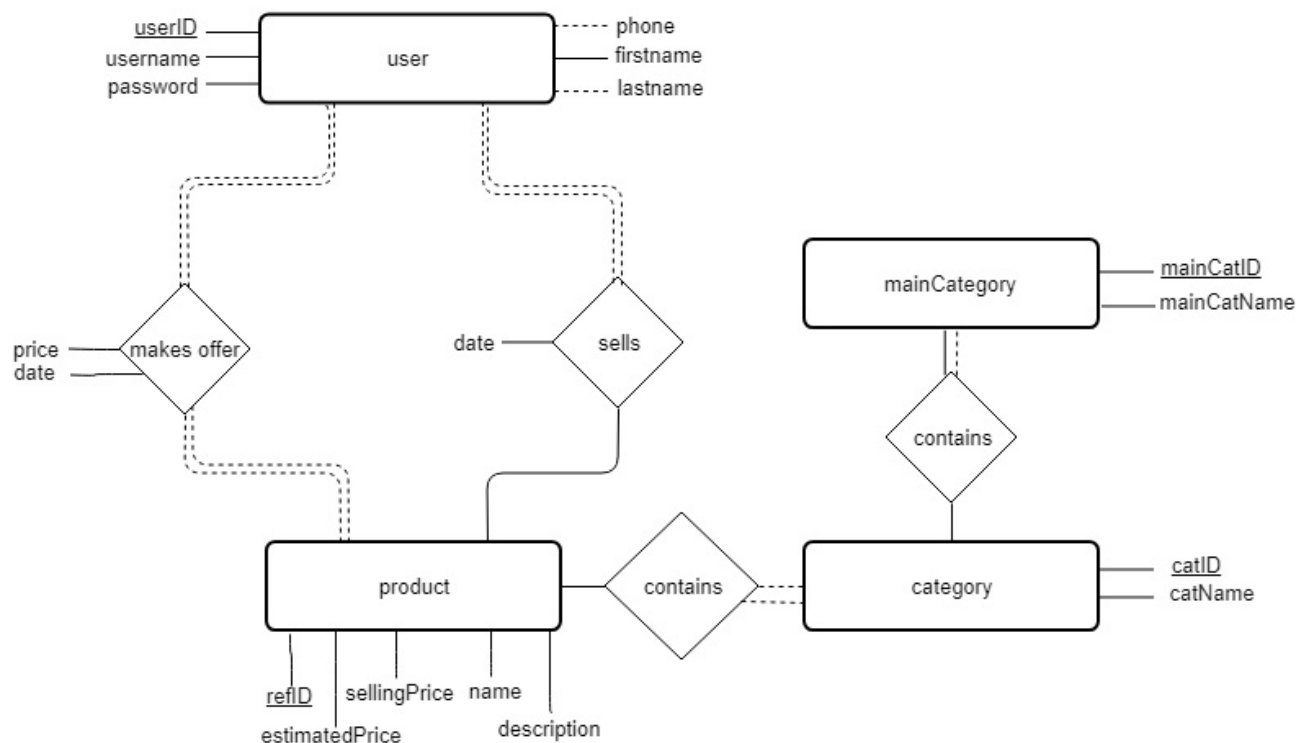
Jonathan Caspar (20059041) - Jules Cohen (20032498)

Jean-François Blanchette (20030091) - Tanahel Huot Roberge (20096767)

Sommaire :

1. Diagramme Entité-Association
2. Modèle relationnel
3. Définition de la base de données
 - i. Explications des choix d'implémentation
 - ii. Fonctions
4. Ensemble des requêtes en SQL et explications des résultats attendus
5. Guide utilisateur

1. Diagramme Entité-Association



Tout d'abord, l'application comporte des utilisateurs inscrits `users` qui doivent déclarer leur **nom d'utilisateur, mot de passe et prénom**. Pour des fins de confidentialité, **les numéros de téléphone et nom de famille** ne sont pas obligatoires. Lors de la création de l'utilisateur, un **numéro d'identification unique** lui sera attribué automatiquement par la base de données.

Une fois enregistré, un utilisateur peut vendre aucun ou plusieurs produits (0 : n) ou faire aucune ou plusieurs offres (0 : n). Quant aux produits, ils peuvent recevoir aucune ou plusieurs (0 : n) offres, n'appartenir qu'à une seule catégorie (1 : 1) et n'avoir qu'un seul vendeur (1 : 1).

Lorsqu'il vend, l'utilisateur crée un nouvel enregistrement dans `products` comprenant le **nom de l'objet**, une **description**, un **prix** et sa **catégorie**. Ensuite, l'utilisateur recevra le **prix estimé** par l'expert et pourra accepter ou refuser de vendre au prix indiqué. Lorsque l'offre de l'expert sera acceptée, le produit se verra attribué la **date de mise en vente**, un **identifiant unique** et **#id vendeur** et **#id catégorie de l'objet**.

Lorsque l'utilisateur fait une offre sur un produit existant, **l'id unique de l'offre**, le **prix offert**, **#id produit**, **#id acheteur** et la **date** seront ajoutés dans la table `offers`. Un produit vendu sera enregistré dans la table `soldProducts` contenant un **id unique**, son **nom**, **description**, **#id du vendeur**, **#id catégorie du produit**, **#id de l'acheteur**, **prix estimé**, **prix de vente**, **prix vendu** et la **date de la transaction**.

Par ailleurs, les catégories principales `mainCategories` sont référencées par leur **id unique** et leur **nom**. Elles contiennent une ou plusieurs (1 : n) sous-catégories qui contiennent, à leur tour, ont un **id unique**, un **nom** et **#idMaincategory**.

Les sous-catégories `categories` ne peuvent avoir qu'une seule (1 : 1) catégorie principale et peuvent contenir aucun ou plusieurs produits (0 : n).

Légende

- #Clé étrangère
- Clé principale
- Attribut d'une table

2. Modèle relationnel

- User(**userID**, username, password, firstname, lastname, phone)
- Product(**refID**, name, description, #sellerID, #categoryID, estimatedPrice, sellingPrice, date)
- Offer(**offerID**, #buyerID, #productID, price, date)
- MainCategory(**mainCatID**, mainCatName)
- Category(**catID**, #mainCatID, catName)
- SoldProducts(**id**, name, description, #sellerID, #buyerID, #categoryID, estimatedPrice, sellingPrice, soldPrice, dateTransaction)

SoldProducts est une table de **log** conservant l'historique des produits vendus.

3. Définition de la base de données (DDL.sql)

Explications des choix d'implémentation :

Nous avons décidé de représenter les `acheteurs` et les `vendeurs` en une seule entité (`user`) ayant un ID arbitraire comme clé primaire. Les experts n'ont pas été représentés car nous ne jugeons pas cela nécessaire (une simple fenêtre suivant la mise en vente suffit).

Les `produits` ont comme clé primaire standard : un ID généré et ont tous une référence (clé étrangère) vers l'entité du vendeur et de sa catégorie.

Les `catégories` sont classés par `catégories principales` (`maincategories`), et chaque produit fait partie d'une seule catégorie. De plus, on force l'unicité de la paire (`catname` et `maincatid`) pour éviter les doublons au sein d'une même catégorie principale. (par

exemple : une catégorie "Autres" pourrait exister plusieurs fois mais pour des catégories principales différentes seulement.)

Les `offres` sont représentés de la même façon que les produits, elles sont identifiées par un ID généré, et ont tous une référence (clé étrangère) vers l'entité de l'acheteur (celui qui a fait l'offre) et le produit ciblé par l'offre.

Nous avons choisi d'ajouter un `ON DELETE CASCADE` à :

```
FOREIGN KEY (productid) REFERENCES products(refid) ON DELETE CASCADE,
```

Afin de faciliter la suppression future des offres pour lesquelles le produit associé n'est plus présent dans la base de données. Ainsi, lorsqu'un produit est supprimé (vente terminée, annulée, etc...) toutes les offres qui y faisaient référence seront aussi supprimées.

La dernière table `soldproducts` nous sert à garder une trace des informations sur les produits vendus.

Fonctions :

- Retourne si oui ou non, un mot de passe correspond à l'utilisateur ayant un ID spécifié

```
CREATE OR REPLACE FUNCTION check_password(id character varying, pwd character varying)
RETURNS boolean
LANGUAGE plpgsql
AS $function$
    DECLARE
        passed BOOLEAN;
BEGIN
    SELECT (password = pwd) INTO passed FROM users WHERE username = id;
    RETURN passed;
END; $function$
```

- Retourne le nom d'une catégorie en fonction de son id

```
CREATE OR REPLACE FUNCTION getcategoriesname(id integer)
RETURNS character varying
LANGUAGE plpgsql
AS $function$
    DECLARE
        name varchar(40);
BEGIN
    name := (SELECT catname FROM categories WHERE catid = id);
    RETURN name;
END; $function$
```

- Retourne le nom d'une catégorie principale en fonction de son id

```
CREATE OR REPLACE FUNCTION getmaincategoriesname(id integer)
RETURNS character varying
LANGUAGE plpgsql
AS $function$
    DECLARE
        name varchar(40);
BEGIN
    name := (SELECT catname FROM maincategories WHERE catid = id);
    RETURN name;
END; $function$
```

- Récupérer le nom et prénom concaténé selon l'id fourni :

```
CREATE OR REPLACE FUNCTION getUserFullName(id integer)
  RETURNS varchar(60) AS $$
  DECLARE
    userfullname varchar(60);
BEGIN
  userfullname := (SELECT CONCAT(firstname, ' ', lastname) FROM users WHERE userid = id);
  RETURN userfullname;
END; $$
LANGUAGE plpgsql;
```

- Comptabilise le nombre d'offres associées au produit en vente (avec refid)

```
CREATE OR REPLACE FUNCTION getOffersCount(refid integer)
  RETURNS integer AS $$
  DECLARE
    offercount integer;
BEGIN
  offercount := (SELECT COUNT(*) FROM offers WHERE productid = refid);
  RETURN offercount;
END; $$
LANGUAGE plpgsql;
```

- Retourne le montant de l'offre la plus élevée pour un produit en vente (avec refid)

```
CREATE OR REPLACE FUNCTION getMaxOfferValue(refid integer)
  RETURNS NUMERIC(10, 2) AS $$
  DECLARE
    offercount NUMERIC(10, 2);
BEGIN
  offercount := (SELECT COALESCE(MAX(price), 0) FROM offers WHERE productid = refid);
  RETURN offercount;
END; $$
LANGUAGE plpgsql;
```

4. L'ensemble des requêtes en SQL et explications des résultats attendus (LMD.sql)

A moins de mentionner un autre id d'utilisateur, on suppose pour les requêtes suivantes que l'utilisateur connecté a un id = 18 :

1) Catalogue

- Afficher les catégories principales sur la colonne gauche :

```
SELECT * FROM maincategories ORDER BY maincatname;
```

Pour chaque catégorie principale : récupérer les sous-catégories et les ajouter dans la colonne de gauche (exemple avec Meubles : id = 6) :

```
SELECT catid, catname FROM categories WHERE maincatid = 6 ORDER BY catname;
```

- En cliquant sur une catégorie : afficher tous les produits (hormis ceux vendus par l'utilisateur lui même)
Exemple avec toutes catégories confondues, on applique un filtre suivant : prix annoncé compris entre 100\$ et 250\$, prix offert compris entre 100\$ et 250\$, date de publication comprise entre le 08-04-2019 et le 10-04-2019

```
WITH allProducts AS
(SELECT refid, name, description, sellingprice, getUserFullName(sellerid) AS sellername,
date, getMaxOfferValue(refid) AS maxoffer, categoryid, estimatedprice
FROM products WHERE
sellingPrice >= 100 AND sellingprice <= 250.0
AND date >='2019-04-08' AND date <='2019-04-10'
AND sellerid <> 18 )

SELECT refid, name, description, sellingprice, sellername, date, maxoffer, catname, date, estimatedprice
FROM allProducts JOIN categories ON categoryid = catid
WHERE maxoffer >= 100.0 AND maxoffer <= 250.0;
```

2) Mes annonces

- Afficher les produits en vente par l'utilisateur (id=18) en précisant le montant de l'offre maximale pour chaque produit

```
SELECT getOffersCount(refid) as nboffers, refid, name, description ,sellingprice,
getUserFullName(sellerid) AS sellername, date, getMaxOfferValue(refid) AS maxoffer,
estimatedprice, categoryid
FROM products WHERE sellerid = 18;
```

- Afficher les offres liées à l'objet pour un utilisateur donné (exemple userid = 18)

```
SELECT offerid, getUserFullName(buyerid) AS buyername, buyerid , productid, name, sellingprice, price ,estimatedprice, offers.
```



- Afficher les offres liées à l'objet sélectionné pour un utilisateur donné (exemple userid = 18, refid = 34)

```
SELECT offerid, getUserFullName(buyerid) AS buyername, buyerid , productid, name, sellingprice, price ,estimatedprice, offers.
```



- Retourne la liste détaillée des produits déjà vendus par l'utilisateur (pour lesquels il a accepté une offre)

```
SELECT name, getUserFullName(sellerid) AS sellername, getUserFullName(buyerid) AS buyername ,soldprice, datetransaction
FROM soldproducts WHERE sellerid = 18;
```

- Accepter l'offre de l'acheteur (id=12) de 125.5\$ sur le produit d'id = 34 vendu par le vendeur d'id=18 . On insère le produit vendu dans la table **soldproducts** et on supprime le produit de la table **products** (ainsi que ses autres offres grâce au ON DELETE CASCADE)

```
INSERT INTO soldproducts (name, description, sellerid, buyerid, categoryid, estimatedprice, sellingprice, soldprice)
SELECT name, description, '18', '12', categoryid, estimatedprice, sellingprice, '125.5' FROM products WHERE refid = 34;
```

3) Mes achats

- Retourne la liste détaillée des offres actives de l'utilisateur connecté

```
SELECT name, getUserFullName(sellerid) AS sellername, sellingprice, price ,offers.date AS date0
FROM products JOIN offers ON productid = refid WHERE buyerid = 18;
```

- Retourne la liste détaillée des produits déjà achetés par l'utilisateur (pour lesquels son offre a été acceptée)

```
SELECT name, getUserFullName(sellerid) AS sellername, getUserFullName(buyerid) AS buyername ,soldprice, datetransaction
FROM soldproducts WHERE buyerid = 18;
```

4) Autres requêtes spécifiques

- Trouve les produits les plus en demande (ceux avec le plus d'offres) par mainCategories

```
WITH nombreOffreParProduit AS
  (SELECT count(*) AS nbrOffre, productid
   FROM offers GROUP BY productid),
mainCatWithChildren AS
  (SELECT catid, maincatid, maincatname
   FROM categories NATURAL JOIN maincategories),
productsWithNbrOffers AS
  (SELECT *
   FROM products JOIN nombreOffreParProduit
    ON refid = productid),
mainCatWithProduct AS
  (SELECT maincatname, refid, name, description, sellerid,
   categoryid, estimatedprice, sellingprice, nbrOffre
   FROM productsWithNbrOffers JOIN mainCatWithChildren ON catid = categoryid),
maxByMainCategory AS
  (SELECT maincatname, MAX(ALL nbrOffre) as nbrOffre
   FROM mainCatWithProduct
   GROUP BY maincatname)
SELECT *
FROM mainCatWithProduct NATURAL JOIN maxByMainCategory;
```

5. Guide utilisateur

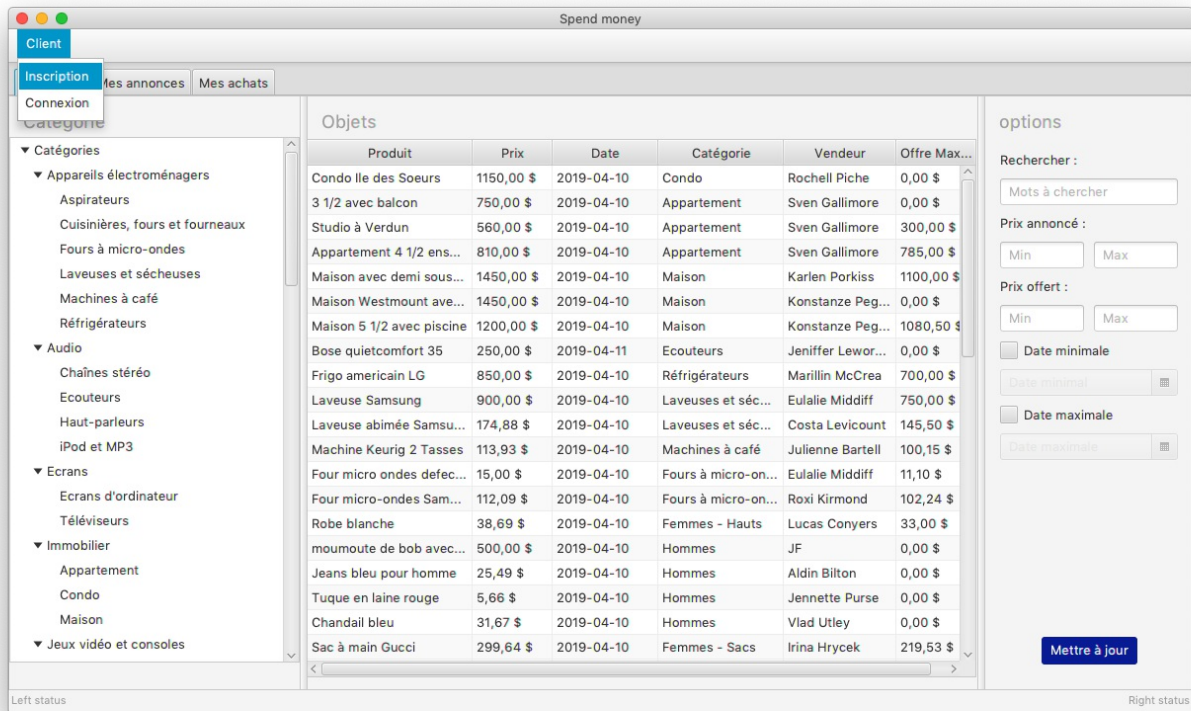
Usage général

Dans l'onglet principal (catalogue), l'interface principale de l'application est divisée en trois boîtes : **Catégories** , **Objets** et **Options** . L'utilisateur peut sélectionner la catégorie générale d'un objet qu'il recherche pour filtrer les objets en vente. Les catégories sont ainsi sous-divisées en catégories plus précises pour simplifier et alléger la recherche de l'objet désiré.

Les objets en vente sont indiqués dans la boîte **objets** et sont filtrables selon le choix de l'utilisateur. Donc, pour faire une recherche en ordre alphabétique, l'utilisateur n'a qu'à cliquer sur **Produit** pour que les objets soient dans l'ordre approprié. Il en va de même pour le reste des colonnes.

Pour afficher les détails d'une annonce, il suffit de double-cliquer sur celle qui intéresse l'utilisateur. Une nouvelle fenêtre s'ouvrira avec les détails et les commentaires de l'article. Il sera possible de faire une offre sur l'objet que lorsque l'utilisateur sera inscrit dans la base de données (voir volet **Inscription au site**).

La boîte **options** permet un filtrage encore plus précis pour l'utilisateur. Il est possible d'entrer les prix minimum et maximum qu'il désire payer, ainsi que les dates d'affichage d'un produit en appuyant sur **Mettre à jour**.



Inscription et connexion au site

Pour accéder aux fonctionnalités de vente, l'utilisateur doit être inscrit dans la base de données du site et se connecter à l'application. En premier lieu, il faut s'inscrire en cliquant sur l'onglet **Client** et sélectionner **Inscription**. Une nouvelle fenêtre s'ouvrira en invitant l'utilisateur à entrer ses informations personnelles et à choisir un nom d'utilisateur avec lequel il sera identifié sur l'application. Le nom et le numéro de téléphone sont facultatifs.

Inscription

INSCRIPTION

Nom d'utilisateur :

Mot de passe:

Nom :

Prénom :

Téléphone :

format : 123-456-7899

S'inscrire

Lorsque l'inscription est réussie, l'utilisateur a, désormais, accès à toutes les fonctionnalités du site. Cliquez sur connexion, entrez vos informations d'utilisateur et cliquez sur [Se connecter](#).

AUTHENTIFICATION

Ouvrir une session

Nom d'utilisateur :

Mot de passe :

Se Connecter

Faire une offre/acheter un produit

Pour faire une offre sur un produit en vente, il suffit de suivre la procédure indiquée dans la section [Usage général](#) et indiquer le prix prêt à offrir dans l'espace prévu à cet effet. Une offre supérieure au prix proposé par le vendeur conclura automatiquement la vente du produit. Une offre inférieure au prix de vente enverra une offre au vendeur qui a le choix d'accepter ou non le prix proposé par l'utilisateur. Lorsque la vente est concluante, une fenêtre confirmant l'achat sera affichée à l'utilisateur.

Offre

Playstation 4

Vendeur :
Jules

Date de création :
2019-04-11

Catégorie :
Sony PlayStation 4

Prix demandé :
250,00 \$

Prix maximum actu...
0,00 \$

Description

avec 2 jeux

Offre

Offrir

Spend money

Client

Catalogue

Mes annonces

Mes achats

Catégorie

Nintendo Wii

Sony PlayStation 3

Sony PlayStation 4

XBOX 360

XBOX One

▼ Livres

Bandes dessinées

Essais

Magazines

Manuels

Ouvrages de fiction

▼ Maison - Intérieur

Décoration intérieure et accessoires

Eclairage intérieur et plafonniers

Literie

Rangement et organisation

Tapis et moquettes

Vaisselle et articles de cuisine

▼ Meubles

Chaises, Fauteuils

Commodos et armoires

Offre

Playstation 4

Vendeur :
Jules

Date de création :
2019-04-11

Catégorie :
Sony PlayStation 4

Prix demandé :
250,00 \$

Prix maximum actu...
0,00 \$

Description

avec 2 jeux

Votre offre de : 190,00 \$ pour Playstation 4 a été effectué avec succès.

L'offre a été ajoutée à la liste.

OK

190

Offrir

options

Rechercher :
Mots à chercher

Prix annoncé :
Min Max

Prix offert :
Min Max

☐ Date minimale
Date minimal

☐ Date maximale
Date maximale

Mettre à jour

Left status

Right status

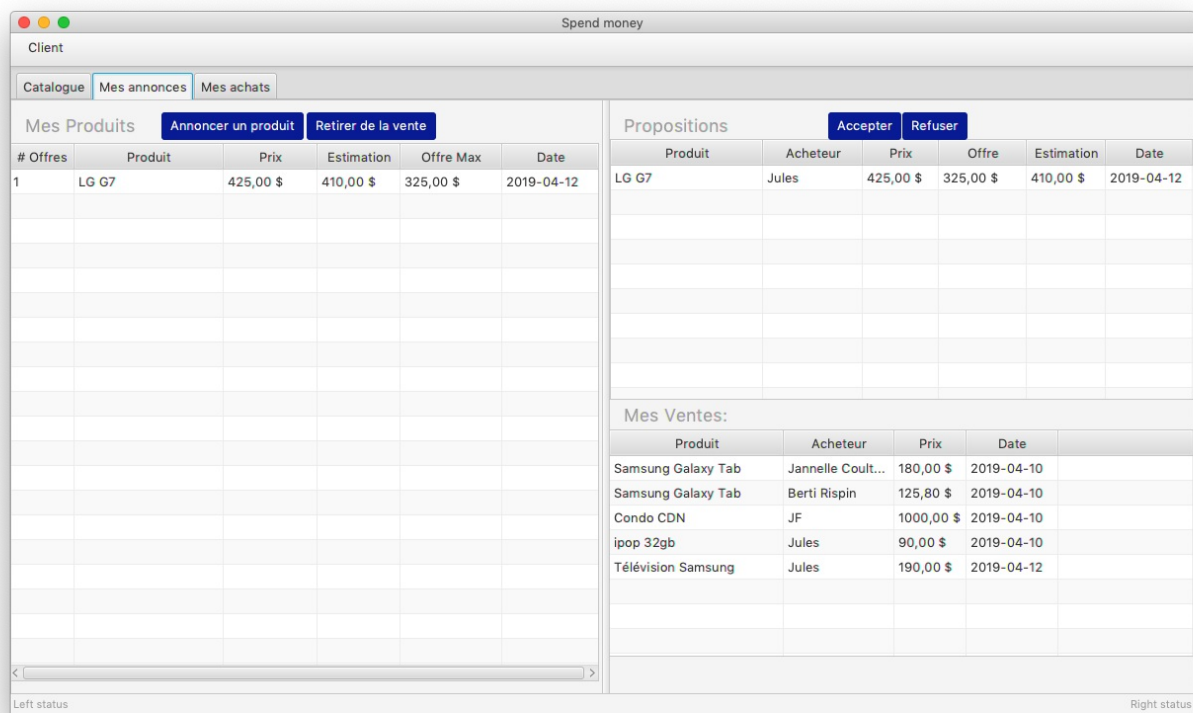
Vendre/Supprimer un produit

À la suite de la connexion à l'application, il faut cliquer sur l'onglet **Mes annonces** de l'interface principal pour accéder aux options de vente. La boîte **Mes Produits** indique les produits mis en vente par l'utilisateur et la boîte **Propositions** affiche les offres reçues par différents clients potentiels.

Pour mettre un objet en vente, cliquer sur **Annoncer un produit**. Une nouvelle fenêtre s'ouvre où les informations de l'objet devront être entrées. Une fois les espaces comblés, l'utilisateur recevra un prix proposé par l'expert selon le marché. Libre à l'utilisateur d'accepter ou non la proposition. Si le prix de l'expert est accepté, l'objet sera mis en vente au prix proposé par ce dernier. Sinon, l'objet ne sera pas mis en vente.

Pour accepter une offre reçue d'un acheteur, il suffit de sélectionner celle-ci dans la boîte **Propositions** et cliquer sur **Accepter**. De la même façon, pour refuser une offre, il suffit de cliquer sur **Refuser**. Les détails de la vente seront indiqués dans la boîte **Mes ventes**. Une offre acceptée entraînera un refus automatique de toutes les autres offres en attente du même produit.

Pour annuler la mise en vente d'un objet, cliquer sur **Retirer de la vente**. L'objet sera retiré de la liste des produits annoncés, ainsi que toutes les offres reçues le concernant.



Nouvelle annonce

ANNONCER VOTRE PRODUIT

Nom de l'objet :

Description :

Prix :

Catégories principales :

Catégories :

Soumettre

□

Achats effectués

L'onglet `Mes achats` de l'interface principal permet de conserver l'historique des achats de l'utilisateur. Ils y sont indiqués en ordre chronologique et contiennent les informations pertinentes de l'achat. Dans la boîte de gauche, les offres en suspens y sont affichées. Une offre en suspens n'est pas refusée, mais bien en attente de confirmation par le vendeur.

