

Programmation  
Systeme &  
Reseaux

# Mini-Projet

## Vente aux Enchères

Réaliser par

**Wala SIDHOM**

**Malek SMIDA**

**Mohamed ABDELHEDI  
NOUMI**

# Sommaire

<b>I.</b>	<b>Présentation du projet .....</b>	<b>3</b>
1-	Introduction .....	3
2-	Objectif .....	4
3-	Fonctionnalité .....	4
4-	Données manipulées .....	5
5-	Langage de la programmation utilisées .....	6
<b>II.</b>	<b>Réalisation du travail .....</b>	<b>6</b>
1-	Gestion des fichiers .....	6
2-	Modèle d'architecture client / serveur .....	7
3-	Echange client / serveur .....	7
4-	La synchronisation des requêtes .....	8
5-	Captures de l'application .....	8
<b>III.</b>	<b>L'organisation du travail .....</b>	<b>11</b>
1-	La méthodologie .....	11
2-	Les choix du travail .....	11
3-	Les difficultés du travail .....	12
4-	Les acquis .....	12

# I. Présentation du projet

## 1- Introduction

L'objectif de ce projet est de mettre au point un système centralisé de vente aux enchères d'objets. En effet, la vente aux enchères est l'une des plus anciennes techniques de vente. Actuellement, de nombreux particuliers et professionnels font appel à ce système de vente pour se débarrasser de certains objets afin de pouvoir obtenir de la liquidité pour financer un nouveau projet ou tout simplement de rembourser quelques dettes. Les ventes aux Enchères sont ouvertes à toute personne majeure.

Préalablement, le commissaire-priseur se charge d'estimer le prix de départ de chaque bien. Par la suite, la vente aux enchères sera annoncée afin d'attirer le plus grand nombre d'acheteurs potentiels. Le jour J, le commissaire-priseur présentera tous les objets disponibles à la vente et annoncera la mise à prix de départ. Par la suite, les personnes désireuses d'acquérir un objet devront enchérir le bien et proposer chacun son prix. La vente se clôturera dès lors que le commissaire-priseur a annoncé le dernier prix et qu'aucun autre acheteur n'a proposé un nouveau prix au bout de 30 secondes. Le commissaire-priseur abat alors son marteau et le dernier acheteur ayant énoncé le prix le plus élevé se verra attribué l'objet.

Par la suite, il devra aller récupérer son bien et le payer immédiatement, car il n'existe pas de délai de réflexion ou de rétractation dans les ventes aux enchères. De plus, le prix de vente sera augmenté d'environ 20 % selon le montant des frais à payer en supplément.

## 2- Objectif

Ce projet de développement a pour objectif la mise en place d'un système informatique. Ce système doit fournir un ensemble de fonctions et doit être architecturé selon le modèle client-serveur. Il doit gérer des informations multiples de manière persistante et doit utiliser un protocole de communication bien défini entre les clients et le serveur et en considérant la synchronisation des requêtes.

## 3- Fonctionnalités

Le système doit fournir les fonctionnalités suivantes :

Pour le commissaire-priseur :

- **Consulter la liste des biens** : en donnant la référence d'un bien, on doit pouvoir récupérer les informations le concernant (prix de départ, dernier prix, état et l'identificateur de son acheteur s'il est vendu).
- **Consulter la facture d'un acheteur** : il doit être possible de voir la facture correspondant à un acheteur en précisant son identificateur.
- **Consulter l'historique des propositions** : il doit être possible de voir le contenu de l'historique des propositions sur un bien donné.

Pour l'acheteur :

- **Acheter un bien** : un acheteur doit pouvoir acheter un bien disponible. Il doit être capable de proposer une valeur pour un bien mis aux enchères.
- **Recevoir une facture** : un acheteur doit recevoir une facture à chaque fois qu'il a achevé sa participation aux enchères.

Le serveur annonce le lancement d'une vente aux enchères d'un bien donné et donc il doit spécifier sa référence et son prix de départ. Selon la proposition de l'acheteur, le poste client préparera une requête à envoyer au serveur.

La requête sera envoyée au serveur et le client se mettra en attente de la réponse.

Le serveur réceptionnera la requête et la traitera pour comprendre la demande du client.

Il effectuera ensuite le traitement associé,

Le serveur enverra le résultat de ce traitement aux différents postes clients si le prix du bien a été mis à jour.

Le client réceptionnera le résultat et pourra enchaîner sur une nouvelle requête.

## 4- Données manipulées

### a- Données des objets mis aux enchères

- La référence précise de l'objet
- Le prix de départ
- Le dernier prix qui reflète à chaque instant la dernière valeur proposée par un acheteur
- L'état qui indique si le bien a été vendu ou non
- L'acheteur qui a emporté l'objet si la vente est conclue

### b- Données des séances de vente

- Identificateur de l'acheteur
- La valeur de la proposition
- Résultat : succès/échec

### c- Données des factures

- L'identificateur de l'acheteur ;

La somme totale à payer par cet acheteur.

## 5- Langage de la programmation utilisée

C'est un langage de programmation objet interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.



## II. Réalisation du travail

### 1- Gestion fichiers

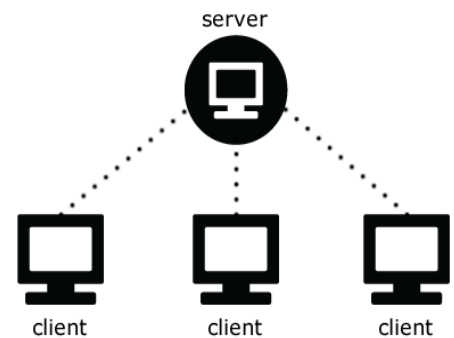
Afin de sauvegarder les données des biens, factures et historiques nous optons à utiliser les fichiers texte qui étaient utiles pour la consultation et la mise à jour des données. Les fonctions qui ont été utilisées sont l'ouverture « `.open()` » avec les différents modes d'ouvertures « `read` », lecture « `.read()` / `.readLines()` » et l'écriture « `.write()` ».

La concurrence d'accès au fichier exige l'emploi d'un mécanisme de synchronisation.

## 2- Modelé d'architecture client / serveur

Dans une relation client/serveur, un programme (le client) demande un service ou une ressource à un autre programme (le serveur).

L'environnement client/serveur désigne un mode de communication entre plusieurs programmes : le client envoie des requêtes et le serveur attend les requêtes des clients et y répondent.



Ainsi chaque application est composée de logiciel serveur et logiciel client pour assurer la communication et le partage des informations.

Pour notre système, nous disposons un serveur et plusieurs clients qui peuvent être des vendeurs ou simples clients.

## 3- Échange client / serveur

La communication a été établie en utilisant les sockets. Un socket est un point de terminaison d'une communication bidirectionnelle, c'est-à-dire entre un client et un serveur en cours d'exécution sur un réseau donné. Les deux sont liés par un même numéro de port de sorte que la couche puisse identifier la demande de partage de données.

Un serveur fonctionne sur une machine bien définie et est lié à un numéro de port spécifique. Le serveur se met simplement à l'écoute d'un client, qui demande une connexion.

Les échanges entre les clients et le serveur doivent suivre un protocole bien défini pour que le serveur comprenne les requêtes des clients et pour que les clients comprennent les résultats renvoyés par le serveur. Le protocole à adopter dans ce cas de projet est TCP et ce pour assurer une communication en mode connecté et mettre en temps réel le contenu des différents fichiers.

#### 4- Synchronisation des requêtes

La synchronisation est une étape très importante parce qu'elle permet d'éviter des résultats erronés. Synchroniser les tâches est très simple à l'aide des threads qui facilitent le déroulement des requêtes sans avoir un blocage. La programmation utilisant des threads est toutefois plus rigoureuse que la programmation séquentielle. En effet, le partage de ressources entre threads permet une communication plus efficace entre les différents threads d'un processus qu'entre deux processus distincts.

#### 5- Captures de l'application

Nous présentons dans cette partie quelques captures d'écran :

L'attente du serveur pour la connexion du client <serveur>

```
===== RESTART: C:\Users\USER\Desktop\sysRapport\Sys\serveur.py =====  
Creation du serveur echo Multi-thread  
Attente de connexion client  
|
```



La connexion du client <Client>

```
===== RESTART: C:\Users\USER\Desktop\sysRapport\Sys\client.py =====
*****

*      Connecté au serveur      *

*****

Warning (from warnings module):
  File "C:\Python27\lib\getpass.py", line 92
    return fallback_getpass(prompt, stream)
GetPassWarning: Can not control echo on the terminal.
Warning: Password input may be echoed.
Tapez le mot de passe :
```

L'affichage du menu <Client>

```
===== RESTART: C:\Users\USER\Desktop\sysRapport\Sys\client.py =====
*****

*      Connecté au serveur      *

*****

Warning (from warnings module):
  File "C:\Python27\lib\getpass.py", line 92
    return fallback_getpass(prompt, stream)
GetPassWarning: Can not control echo on the terminal.
Warning: Password input may be echoed.
Tapez le mot de passe : client
Mot de passe accepté...
Tapez votre Option :

***** MENU *****
*
*  1/ Consulter Un produit      *
*
*  2/ Acheter Un produit        *
*
*  3/ Consulter historique      *
*
*  4/ Consulter facture         *
*
*  5/ exit                      *
*
*****
```

## Exemple de consultation et achat du produit <Client>

Tapez votre Option :

```
***** MENU *****
*
* 1/ Consulter Un produit      *
*
* 2/ Acheter Un produit       *
*
* 3/ Consulter historique     *
*
* 4/ Consulter facture        *
*
* 5/ exit                     *
*
*****
```

1

Donnez la reference de produit : 2

2 0 3 0 1

Tapez votre Option :

- 1) Consulter Un produit
- 2) Acheter Un produit
- 3) Consulter historique
- 4)consulter facture
- 5)exit

2

saisir la reference de produit : 2

saisir prix :500

saisir votre identifiant :1

Tapez votre Option :

- 1) Consulter Un produit
- 2) Acheter Un produit
- 3) Consulter historique
- 4)consulter facture
- 5)exit

### III. L'organisation du travail

#### 1- Méthodologie

Piloter un projet comporte plusieurs dimensions organisationnelles comme l'organisation des tâches du projet, la formation des équipes et le développement des compétences.

Au début l'idée de la réalisation du projet était un peu floue. Avant de commencer à travailler, nous avons effectué des recherches afin de pouvoir trouver des projets similaires qui nous éclaircissent les idées et nous guident.

Ensuite, nous avons défini un plan général des phases de la réalisation (Communication client/serveur via socket / manipulation fichier et maj / synchronisation).

En addition, la présence de tout le groupe et la collaboration était primordiale.

NB : la lecture de l'énoncé de projet était une étape très importante qui accompagne tout le cursus de travail et a joué un rôle très important pour comprendre toute ambiguïté rencontrée.

#### 2- Les choix du travail

Les principaux choix de la réalisation du projet et qui avait un impact positif sont :

- Le choix des membres de l'équipe : la bonne collaboration entre une équipe de travail et la présence d'une ambiance amicale conduit à un travail réussi.
- Les sources d'informations ont été bien déterminé après une bonne recherche, nous avons constaté que les sources qui peuvent nous aider tout au long du projet sont les sites : YouTube, StackOverFlow.

### 3- Les difficultés du travail

Les difficultés rencontrées sont :

- Le passage de la théorie en pratique est une tâche difficile mais avec la programmation système nous avons remarqué combien il y a une large différence vue que c'est un type de programmation qui vise au développement des programmes qui font partie de SE.
- Après 30s de silence si aucun client n'a proposé un autre prix, le produit sera effectué au dernier qui a donné le dernier prix.
- La modification automatique des fichiers.

### 4- Les acquis

Le projet était une opportunité d'appliquer les connaissances vues en cours. Il nous a donné la possibilité :

- De faire une bonne gestion des tâches et du temps.
- De comprendre le modèle de l'architecture client / serveur.
- De réaliser un travail collectif en équipe.
- De programmer avec un nouveau langage « Python ».