

RFC version 2 – Projet Cbay

Lorrain BALBIANI

Manavai TEIKITUHAAHAA

Table des matières

Introduction.....	2
Formalisme.....	2
Connexion au serveur.....	3
Structures de donnée.....	4
Structure des requêtes.....	5
Liste des variables	6
I. Inscription d'un nouvel utilisateur	7
1. Vérification de la disponibilité d'un nom d'utilisateur.....	7
2. Inscription d'un nouvel utilisateur	8
3. Exemple	9
II. Connexion.....	11
1. Connexion d'un utilisateur	11
2. Choix du mode.....	12
3. Exemples.....	13
III. Mode administrateur	15
1. Gestion des utilisateurs.....	15
a) Changement de mot de passe d'un utilisateur	15
b) Suppression d'un utilisateur.....	17
2. Gestion des objets aux enchères.....	18
IV. Mode vendeur.....	20
1. Affichage des ventes effectuées.....	20
2. Gestion des ventes en cours.....	21
3. Ajout d'un nouvel item.....	23
V. Mode Acheteur	24
1. Affichage historique des achats	24
2. Gestion enchères en cours	25
3. Enchère sur un nouvel item	27

Introduction

Dans le cadre d'un projet de programmation en langage C, d'une architecture client/serveur d'enchère, correspondant à notre cours d'Informatique de notre cursus d'ingénieur T.R.I, nous devons produire une RFC relatant les échanges entre ces entités. Cette RFC est la seconde version du Projet de programmation Cbay.

Dans cette version, on propose des améliorations dans les échanges entre le serveur et le client, pour faciliter leurs extractions, leur formatage ou bien leur compréhension par rapport à la dernière RFC.

Cette version comporte aussi des captures d'écran d'une interface en ligne de commande, pour inspirer les développeurs de ce Projet. Pour certaines fonctionnalités, des captures d'écrans n'ont pas été effectué du fait que celles-ci ne sont pas encore opérationnelles sur les applications.

Formalisme

Requête du client : 

Réponse du serveur :

Normal : 

Erreur : 

Les mots dans une requête ou une réponse présentant un « \$ » en début de mot est une variable.

Exemple :

\$USER signifie que le nom d'utilisateur devra être remplacé dans la requête à la position de cette variable.

\$MODE signifie le mode dans lequel un utilisateur est connecté.

En Annexe, vous trouverez les structures définissant les paramètres des clients et du serveur, ainsi qu'une liste répertoriée de toutes les erreurs qui pourront être envoyées par le serveur en fonction des requêtes.

Les codes d'erreur qui seront en fin de RFC, sont au format hexadécimal suivant :

- 0x000
- 0x001

Un utilisateur standard désigne un utilisateur qui n'est pas Administrateur.

Le terme « doit » désigne l'obligation d'une action qui devra être effectuée par les développeurs du client ou du serveur.

Modification par rapport à la dernière version :

Toutes les modifications par rapport à la dernière version de cette RFC seront mises en couleur violet.

Compte aux échanges entre le serveur et le client, elles seront encadré de la même couleur puis suivis d'une justification en couleur violet :

REQUETE ou REPONSE MODIFIE dans cette RFC

Connexion au serveur

La connexion au serveur se fait sur un socket TCP/IP sur un numéro de port décidé par les développeurs de l'application.

Le serveur ne gère qu'une seule connexion à la fois pour faciliter la programmation et pour éviter toute concurrence sur un objet qui causera des erreurs non-prises en compte dans cette RFC.

Structures de donnée

Les structures de donnée citée ci-dessous peuvent être modifiées en conséquence selon les besoins, les fonctionnalités ou les optimisations recherchées par les développeurs.

En bref ces structures ne sont là que pour inspirer les programmeurs.

Voici la structure de donnée minimum définissant un client (utilisateur de l'application cliente) :

```
struct user_t
{
    unique_id_t uid;           // IDENTIFIANT UNIQUE
    char login[21];            // NOM D'UTILISATEUR
    char password[21];         // MOT DE PASSE
    bool admin;                // MODE ADMIN(TRUE or FALSE)
}
```

Voici la structure de donnée minimum définissant un objet :

```
struct object_t
{
    unique_id_t uid;           // IDENTIFIANT UNIQUE DE L'ITEM
    char name[51];             // NOM DE L'OBJET (à l'unité)
    // exemple: pour un lot de XXX: "XXX"

    char category[51];         // CATEGORIE (mobilier, électro,
    // auto, vêtement, livre, )

    char description[1024];     // DESCRIPTION DE L'OBJET OU DU LOT
    // MIS AUX ENCHERES

    char url_image[100];       // URL IMAGE
    float start_price;         // PRIX DE DEPART DE L'ENCHERE
    float temp_price;          // PRIX INTERMEDIAIRES (prix proposé
    // durant l'enchère par les utilisateurs, le prix affiché est celui le
    // plus élevé de la mise)

    float final_price;         // PRIX FINAL DE L'ENCHERE
    int quantity;              // QUANTITE (le prix de l'enchère
    // est pour tout le lot)

    char place[50];            // ADRESSE DE L'OBJETS
    unique_id_t vendeur;       // UID DU VENDEUR
    unique_id_t acheteur;      // UID DE L'ACHETEUR (initialisé à 0
    // / mis à 0 lors de la suppression d'un utilisateur lors d'un enchère)
    time_t expire_time         // TEMPS avant expiration de
    // l'enchère
};
```

Attention, la taille des variables dans les requêtes ou réponse doit être au maximum égale à la taille prévu dans les structures de données de cette RFC. Par exemple, un nom d'utilisateur ne devra pas comporter plus de 20 caractères.

Modification : Un champ a été rajouté à la structure de données (expire_time) pour ajouter une nouvelle fonctionnalité d'expiration des objets vendu aux enchères.

Structure des requêtes

Les requêtes seront tous de la forme suivante :

REQ_MOBILE_DE_LA_REQUÊTE = \$ARGUMENT1 FOR ARGUMENT2 BY ARGUMENT3 \n
--

Les requêtes sont tous délimitées par un mot clé au début et terminées par un « \n ». Le mot clé ne doit pas comporter d'espace entre les mots, mais des tirets (Under-score « _ »). Le séparateur suivant le mot clé de la requête est le signe égal « = » avant d'être suivi par une suite d'argument et de séparateur qui peuvent être :

- Le mot « FOR » ;
- Le mot « BY » ;
- Le signe « + » ;
- Un espace.

Attention, chaque mot de la requête doit être obligatoirement séparé par un espace (sauf le mot clé).

Une requête doit être suivie par une réponse du serveur. La réponse est du même format, c'est-à-dire qu'il doit comporter au minimum un mot clé et doit être délimitée par un « \n » en fin de phrase.

Exemple :

REQ_CONNECTION = \$LOGIN + \$PASSWORD \n
--

 est suivi de

USER_CONNECTED = \$UID \n.

Cette requête correspond par exemple à la connexion d'un utilisateur.

Liste des variables

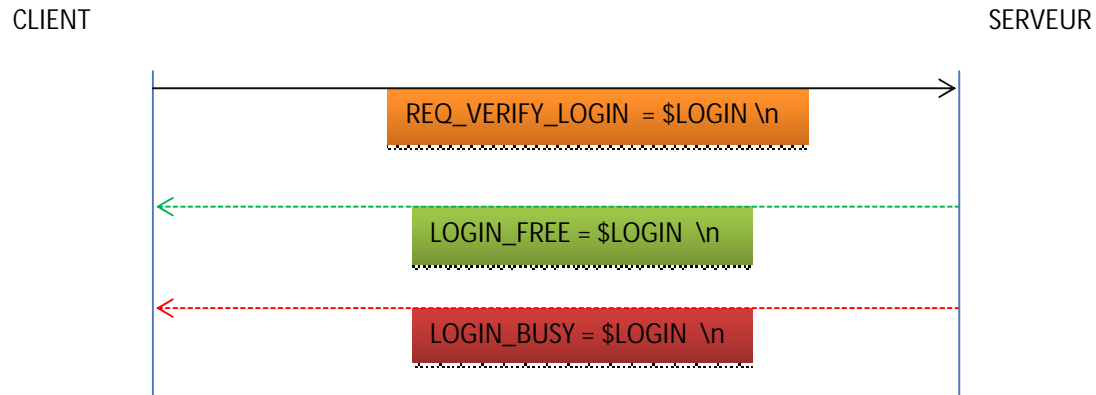
\$LOGIN	nom d'utilisateur envoyé ou reçu.
\$PASSWORD	mot de passe utilisateur qui sera envoyé dans la requête.
\$UID	Identifiant Unique (UID) de l'utilisateur
\$MODE	mode de l'utilisateur (acheteur, vendeur ou administrateur).
\$ADMIN.UID	UID de l'administrateur
\$ADMIN.LOGIN	Nom d'utilisateur de l'administrateur.
\$NEW_PW	Nouveau mot de passe envoyé au serveur pour un utilisateur
\$OPERATION	Opération de va subir un objet ou utilisateur
\$ITEM	Désigne l'objet des enchères. \$ITEM.XXX signifie l'accès aux champs XXX dans la requêtes
\$MSG_ERROR	Code d'erreur renvoyé par le serveur en cas de problème

Plusieurs variables ne sont pas lister dans ce tableau. Elles sont cependant évidentes à comprendre.

I. Inscription d'un nouvel utilisateur

Une fois que le programme client est connecté au serveur, celui-ci aura le choix de pouvoir inscrire un nouvel utilisateur avant l'authentification d'un utilisateur. Cette inscription se fait en deux étapes, la vérification de la disponibilité d'un nom d'utilisateur et son inscription.

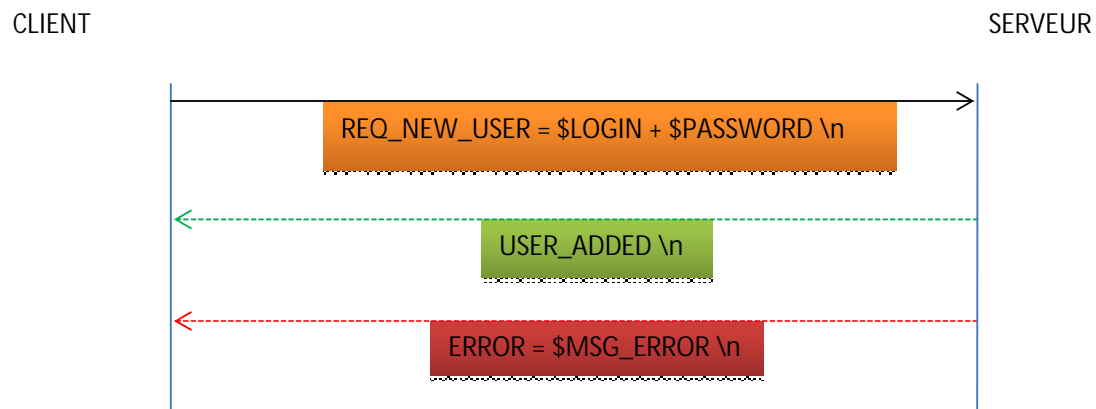
1. Vérification de la disponibilité d'un nom d'utilisateur.



Cette requête permet de vérifier la disponibilité d'un nom d'utilisateur et de prévenir les erreurs lors de l'inscription d'un utilisateur sur le serveur d'enchère.

Si le nom d'utilisateur est disponible, alors le serveur renverra une réponse positive « LOGIN_FREE » ; sinon le serveur renverra une réponse négative « LOGIN_BUSY ».

2. Inscription d'un nouvel utilisateur



Cette requête permet d'inscrire un nouvel utilisateur sur le serveur d'enchère. Il doit comporter un nom d'utilisateur disponible, vérifié au préalable, et un mot de passe qui sont envoyés au serveur.

Une identité unique (UID) est alors créée et correspond à la date d'inscription au format d'heure POSIX (soit le nombre de seconde écoulé depuis le 1er janvier 1970 00:00:00).

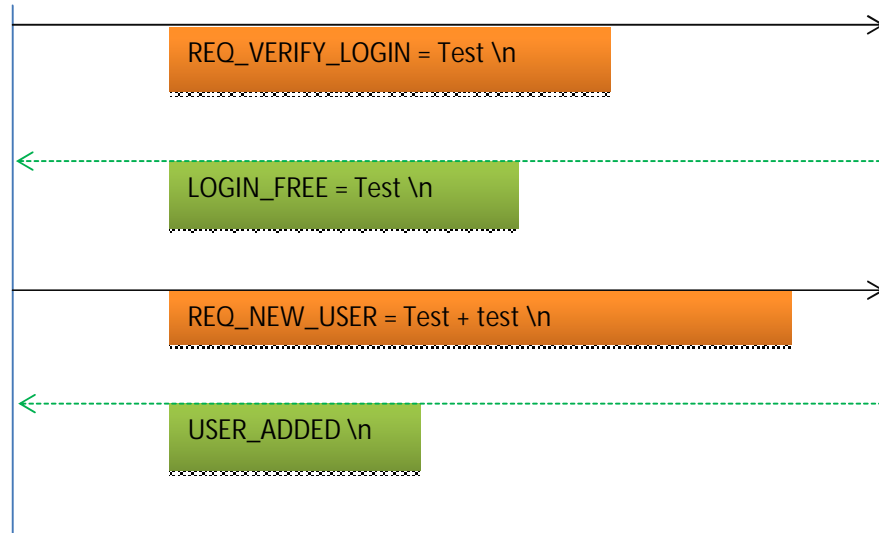
Après traitement par le serveur, celui-ci envoie une réponse positive, « USER_ADDED » ; sinon, il envoie un code d'erreur au format hexadécimal.

3. Exemple

Dans cet exemple, un utilisateur veut vérifier la disponibilité de son nom d'utilisateur avant de tenter une inscription.

Voici les paramètres de l'exemple :

- Nom d'utilisateur : Test
- Mot de passe : Test



On suppose que dans cet exemple que le nom d'utilisateur était libre. Cependant, s'il ne l'est pas, le client doit revérifier la disponibilité d'un nom d'utilisateur alternatif et ce jusqu'à ce qu'il soit validé par le serveur, avant d'envoyer une requête d'inscription. Si le nom d'utilisateur n'est pas libre lors de l'inscription, le serveur renverra une erreur.

Exemple sur une interface en ligne de commande :

L'utilisateur rentre un login puis le serveur répond qu'il est libre. Après avoir rentré un mot de passe pour ce login, le l'utilisateur envoi une requête pour la création d'un nouveau compte utilisateur. Enfin le serveur répond positivement à la requête.

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier Édition Affichage Rechercher Terminal Aide
-----
Bienvenue sur Cbay, Application Client!

Adresse du serveur: localhost

*** CONNEXION ETABLIE AVEC LE SERVEUR ***

Que voulez-vous faire ?
    1 - Se connecter au serveur
    2 - S'inscrire sur le serveur
    0 - Quitter le programme
Choix[1|2|0]: 2

Veuillez rentrer un login: test
Le nom d'utilisateur est libre.
Entrer un mot de passe SVP: test
Votre compte a bien été créé !
```

II. Connexion

1. Connexion d'un utilisateur

\$LOGIN : nom d'utilisateur qui sera envoyé dans la requête.

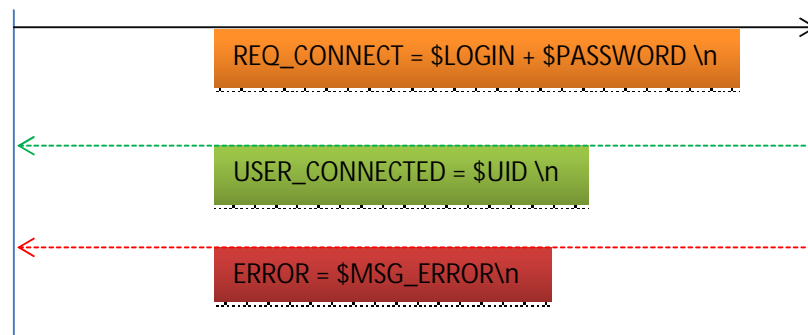
\$PASSWORD : mot de passe utilisateur qui sera envoyé dans la requête.

\$UID : Identifiant unique de l'utilisateur.

\$MSG_ERROR : Code d'erreur renvoyé par le serveur en cas de problème.

CLIENT

SERVEUR



Cette requête permet la connexion d'un utilisateur au serveur d'enchère.

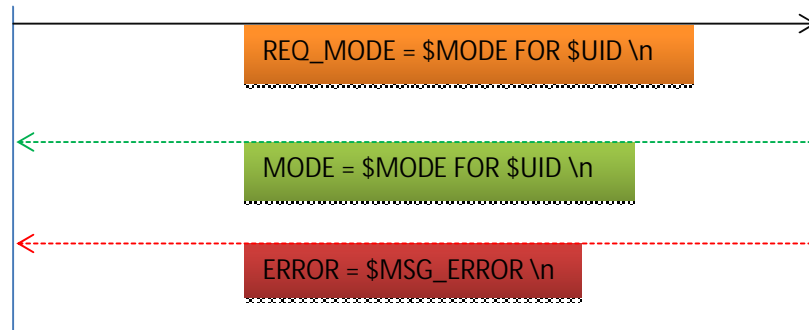
Une requête comportant un nom d'utilisateur disponible et un mot de passe est envoyée au serveur. Si le serveur accepte la connexion après la vérification des identifiants transmis dans la requête, le serveur renvoie l'identifiant unique (UID) de l'utilisateur qui vient de se connecter.

Cette UID permettra l'échange ultérieur entre le client et le serveur des requêtes qui comportera obligatoirement l'UID dans certaine requête confidentiel. Il sera à la charge des développeurs de l'utiliser ou non lors des vérifications.

2. Choix du mode

CLIENT

SERVEUR



Cette requête permet de choisir un mode utilisateur. Il existe trois modes différents :

- Administrateur : permet la gestion des utilisateurs et des objets aux enchères.
- Acheteur : permet à un utilisateur de mettre de nouveaux objets aux enchères, de visualiser ceux actuellement aux enchères ou un historique de ses ventes.
- Vendeur : ce mode permet à un utilisateur d'enchérir sur des objets. Il pourra aussi avoir une visualisation de ses achats.

Attention : Un utilisateur qui vient de s'inscrire sera le seul à pouvoir choisir le mode administrateur et pourra par la suite revenir sur ce mode s'il le souhaite tandis qu'un utilisateur standard n'aura pas le droit de choisir le mode Administrateur même s'il est proposé au niveau du programme client.

Format de \$MODE :

- ADMIN
- BUY
- SELL

Modification : Il n'est plus obligatoire de choisir entre le mode Acheteur ou Vendeur la connexion d'un utilisateur, sauf si les développeurs veulent une trace dans leurs fichiers de journalisation. Il est cependant conseillé d'utiliser cette requête pour le mode Administrateur pour éviter que des utilisateurs non autorisés ne viennent corrompre le fonctionnement du serveur.

3. Exemples

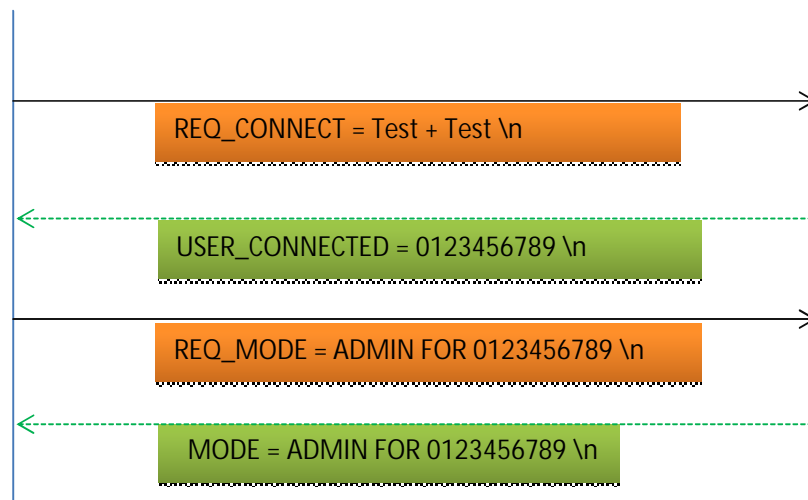
Dans cet exemple, un utilisateur tente de se connecter au serveur et choisit son mode. On supposera que tout se passe bien.

Paramètres de l'exemple :

- Nom d'utilisateur : Test
- Mot de passe : Test
- Mode : ADMIN

Client

Server



Exemple sur une interface en ligne de commande :

L'utilisateur se connecte avec son identifiant et son mot de passe. Le serveur répond positivement à la requête. Dans cet exemple, le client a droit à 4 tentatives avant que la connexion ne soit fermée par le serveur, pour donner l'accès à un autre.

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier Édition Affichage Rechercher Terminal Aide
Votre compte a bien été créé !
Connexion:
    Login: test
    Mot de passe: test
Tentative de connexion 1 ...
Connecté en tant que: test
Choix du mode:
    1 - Mode Acheteur
    2 - Mode Vendeur
    3 - Mode Administrateur
Choix[1|2|3]: █
```

III. Mode administrateur

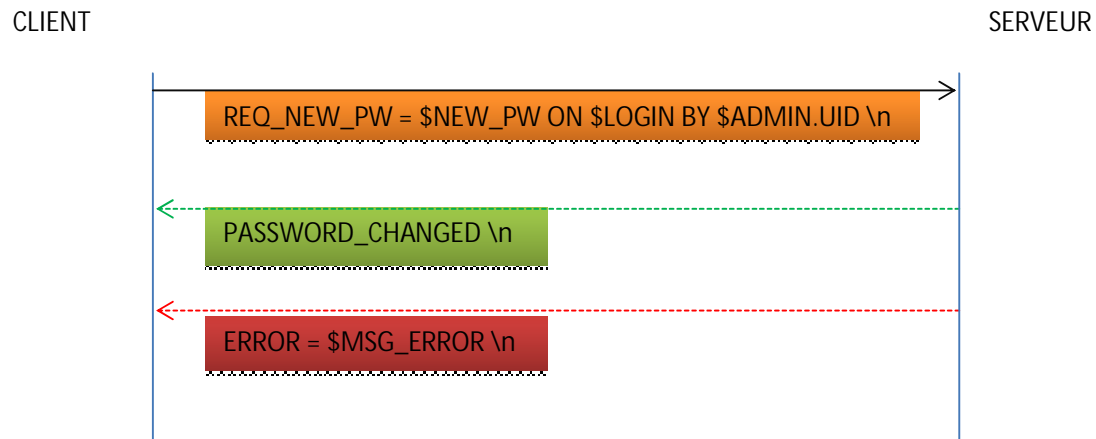
L'Administrateur a les fonctionnalités suivantes :

- Gérer les utilisateurs
- Gérer les ventes (supprimer/annuler une vente)

1. Gestion des utilisateurs

Un administrateur a le droit de soit modifier le mot de passe d'un utilisateur, soit en supprimer un. Lorsqu'un utilisateur est supprimé, tous les objets mis en enchères seront supprimés. Si l'utilisateur a placé des enchères sur un objet alors les enchères seront remises automatiquement à zéro.

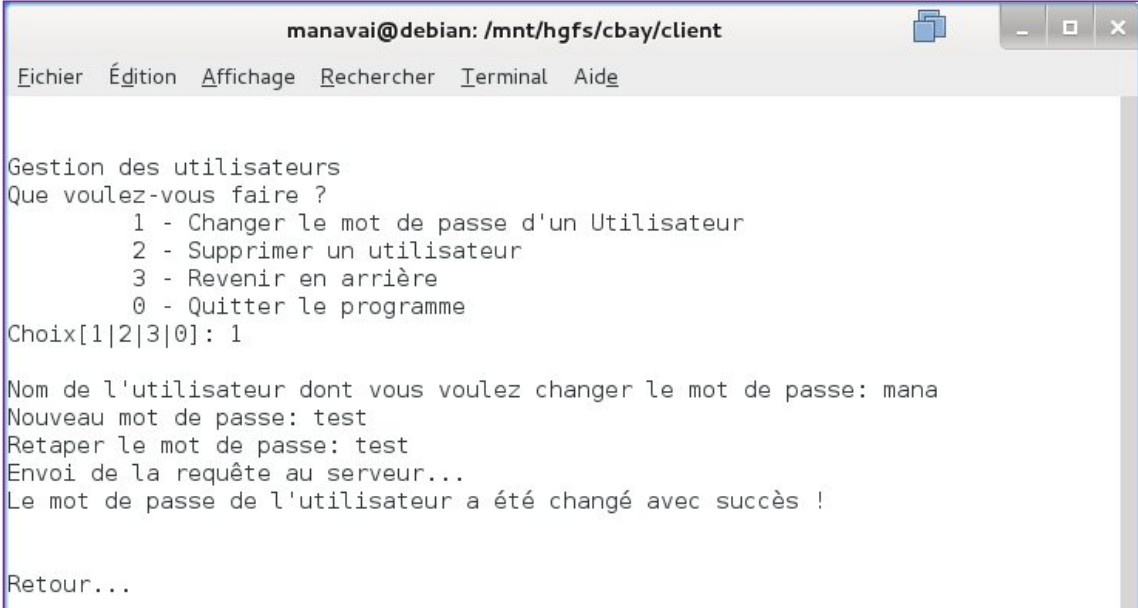
a) Changement de mot de passe d'un utilisateur



Un administrateur qui désire changer le mot de passe d'un utilisateur doit connaître le nom de ce dernier. Ainsi dans la requête de modification du mot de passe, le nouveau mot de passe ainsi que le nom d'utilisateur à modifier seront envoyés. L'UID de l'administrateur ne servant qu'à authentifier la requête.

Si la suppression n'a pas abouti, un message d'erreur au format hexadécimal sera envoyé au client.

Exemple sur une interface en ligne de commande :



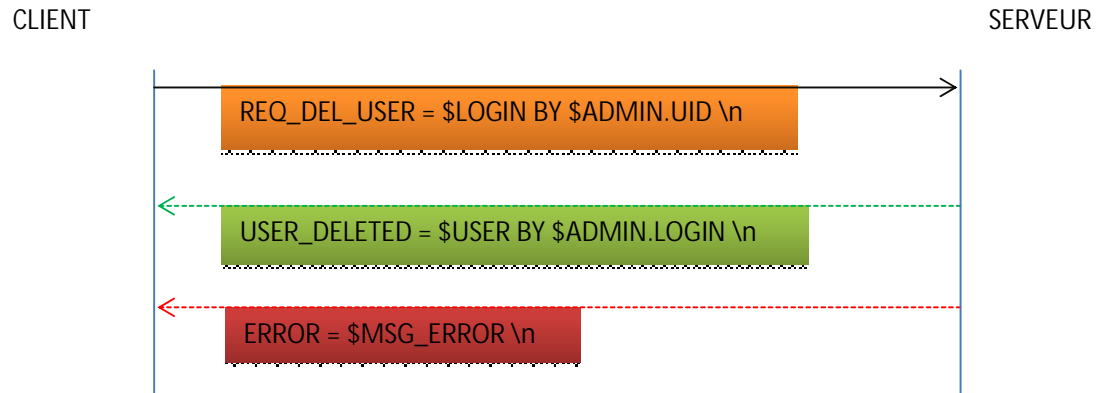
The screenshot shows a terminal window titled "manavai@debian: /mnt/hgfs/cbay/client". The window has a menu bar with "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal content is as follows:

```
Gestion des utilisateurs
Que voulez-vous faire ?
    1 - Changer le mot de passe d'un Utilisateur
    2 - Supprimer un utilisateur
    3 - Revenir en arrière
    0 - Quitter le programme
Choix[1|2|3|0]: 1

Nom de l'utilisateur dont vous voulez changer le mot de passe: mana
Nouveau mot de passe: test
Retaper le mot de passe: test
Envoi de la requête au serveur...
Le mot de passe de l'utilisateur a été changé avec succès !

Retour...
```


b) Suppression d'un utilisateur



Une requête de suppression d'utilisateur ne peut être établie que par un administrateur.

Ainsi seul le nom d'utilisateur qui sera supprimé sera envoyé dans la requête, suivi de l'UID de l'administrateur pour authentifier la requête.

Si la suppression n'a pas abouti, un message d'erreur au format hexadécimal sera envoyé au client.

Exemple sur une interface en ligne de commande :

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
ConnectÃ© en tant que: test
Choix du mode:
    1 - Mode Acheteur
    2 - Mode Vendeur
    3 - Mode Administrateur
Choix[1|2|3]: 3
Que voulez-vous faire ?
    1 - Gérer les utilisateurs
    2 - Gérer les ventes(supprimer/annuler une vente)
    0 - Quitter le programme
Choix[1|2|0]: 1

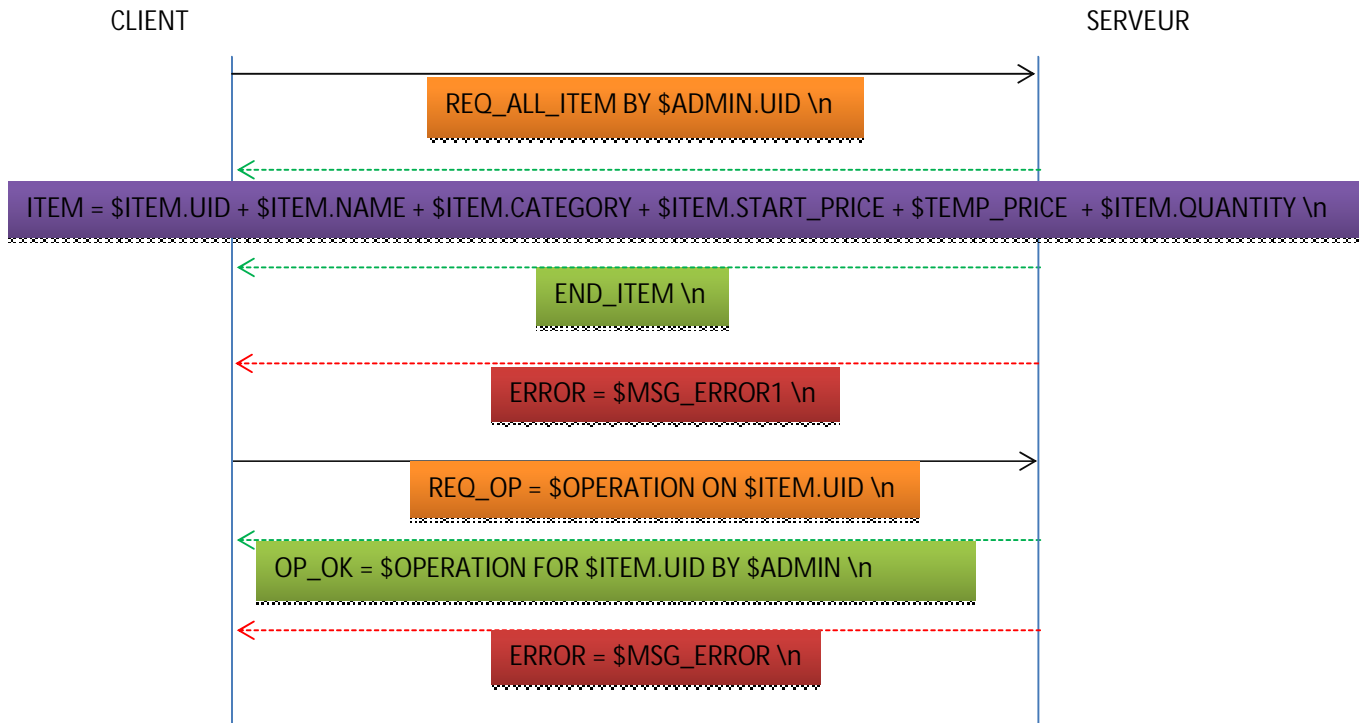
Gestion des utilisateurs
Que voulez-vous faire ?
    1 - Changer le mot de passe d'un Utilisateur
    2 - Supprimer un utilisateur
    3 - Revenir en arrière
    0 - Quitter le programme
Choix[1|2|3|0]: 2

Nom de l'utilisateur que vous voulez supprimer: mana
Envoi de la requête au serveur...
L'utilisateur a été supprimé avec succès !
```

2. Gestion des objets aux enchères

Dans la gestion des objets, un administrateur recevra l'affichage des tous les objets qui sont aux enchères. L'affichage des objets est à la charge des développeurs de l'application cliente.

Le serveur enverra les objets en fonction de leur date de mise aux enchères.



Une fois l'affichage terminé le client administrateur pourra effectuer ce type d'opération sur un objet :

Format \$OPERATION :

CANCEL : Supprimer une enchère en cours.

DELETE : Annuler une enchère, c'est-à-dire qu'elle sera remise à zéro et que toutes les offres ayant été effectuées depuis le début seront supprimées.

Si une opération n'a pas pu aboutir, un message d'erreur au format hexadécimal sera envoyé au client.

Modification : Des séparateur « + » ont été ajouté entre chaque donnée (réponse de type ITEM) pour faciliter l'extraction de celles-ci (avec sscanf () par exemple) au cas où il y a la présence d'espace dans ces données (dans le nom ou la catégorie par exemple).

Exemple sur une interface en ligne de commande :

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier Édition Affichage Rechercher Terminal Aide
3 -      UID:1300202314
        Nom: Test 3
        Catégorie: test
        Prix de départ: 15.000000 Euro
        Dernier enchère: 50.000000 Euro
        Quantité: 10

FIN D'AFFICHAGE
Numéro d'item [1 - 3]: 1
Que voulez-vous faire ?
1 - Supprimer l'enchère
Permet de supprimer une enchère en cours.
L'objet sera supprimé du serveur.

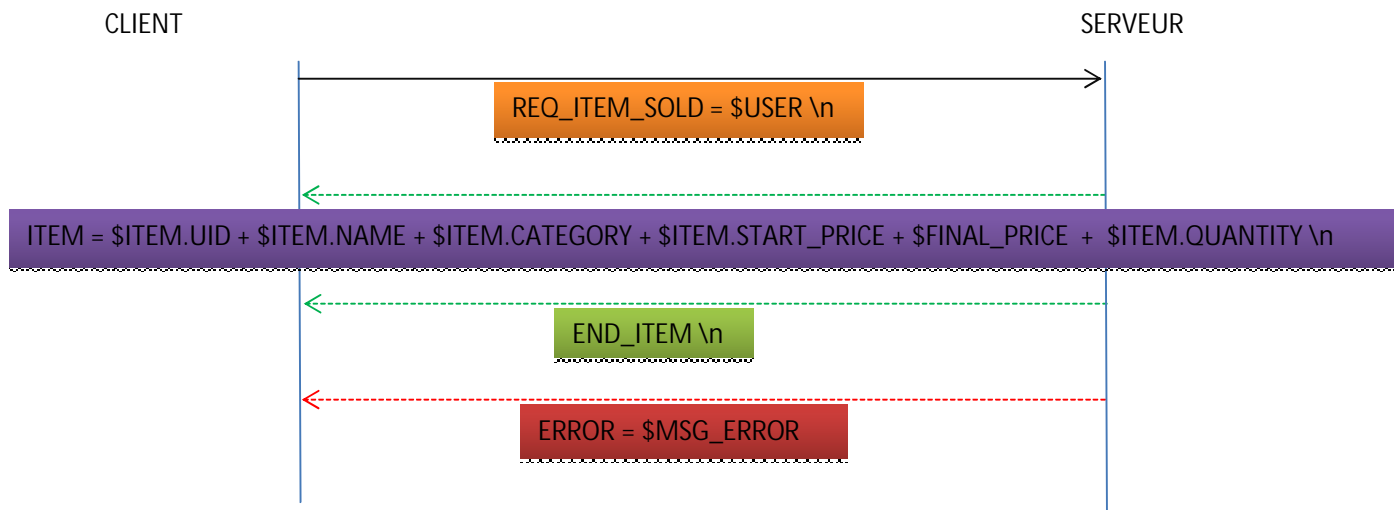
2 - Annuler l'enchère
Permet d'annuler une enchère.
Tous les compteurs de l'objet seront remis à zéro.

0 - Quitter cette fonctionnalité
Choix[1|2|3|0]: 1
L'enchère a bien été supprimé
```

IV. Mode vendeur

Ce mode permet aux utilisateurs de pouvoir mettre des objets aux enchères. Il permet aussi d'afficher les objets que l'utilisateur a mis aux enchères et de pouvoir les gérer. Ce mode permettra à l'utilisateur de consulter son historique de vente.

1. Affichage des ventes effectuées



Lorsqu'un utilisateur désire consulter son historique de vente, le client envoie une requête à l'application serveur avec son identifiant unique. Ensuite un algorithme de recherche effectue une recherche sur les objets en faisant la correspondance entre l'UID du vendeur d'un objet et le prix final (il faut que le prix final soit différent de 0 pour que l'objet soit considéré comme vendu).

Les caractéristiques d'un objet sont envoyées au fur et à mesure de la recherche, et lorsqu'il n'y a plus de correspondance, le serveur envoie au client l'information « END_ITEM ».

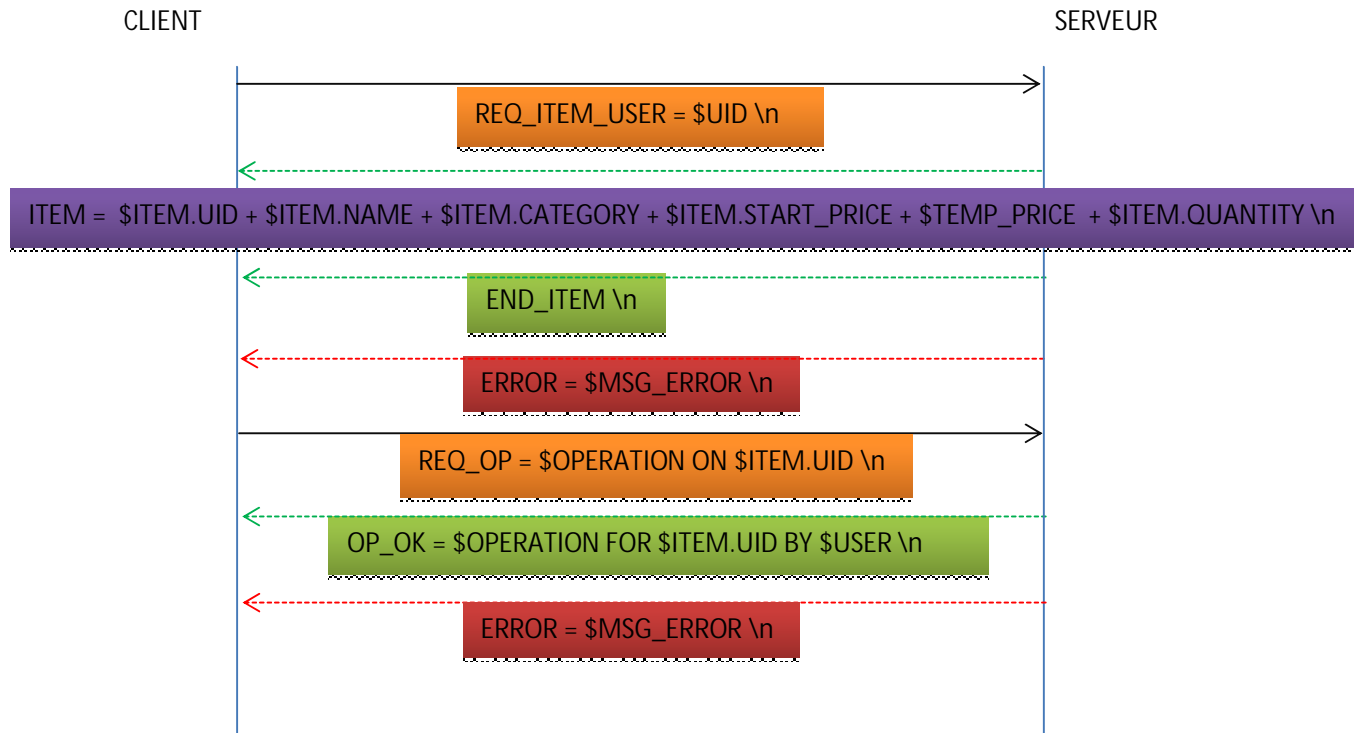
L'affichage des objets est à la charge du développeur de l'application cliente.

Modification : Des séparateurs « + » ont été ajoutés entre chaque donnée (réponse de type ITEM) pour faciliter l'extraction de celles-ci (avec `sscanf()` par exemple) au cas où il y a la présence d'espace dans ces données (dans le nom ou la catégorie par exemple).

2. Gestion des ventes en cours

Le gestionnaire des ventes en cours permet de :

- Afficher les objets en cours d'enchère où vous avez effectué une mise.
- Supprimer une enchère / un objet
- Annuler une enchère.



Lorsque l'utilisateur envoie une requête avec son UID d'utilisateur, le serveur effectue un algorithme de recherche sur l'objet en se basant sur l'UID de l'utilisateur, et le champ `FINAL_PRICE` d'un objet non-vendu qui doit être nul.

A chaque correspondance, le serveur envoie les caractéristiques de l'objet et quand il n'y a plus de correspondance, le serveur envoie l'information « `END_ITEM` ».

La gestion de l'affichage est à la charge du développeur de l'application cliente.

Ensuite l'utilisateur peut sélectionner un objet et y effectuer les opérations citées ci-dessous :

\$OPERATION :

END_BID : Permet de mettre fin à une enchère. La dernière personne à avoir enchéri sera marqué comme acheteur. Cette nouvelle opération donne la possibilité au vendeur de finir une enchère quand il le veut.

DELETE_BID : Permet de supprimer une enchère en cours. L'objet sera supprimé du serveur.

CANCEL_BID : Permet d'annuler une enchère. Tous les compteurs de l'objet seront remis à zéro.

Modification : Des séparateur « + » ont été ajouté entre chaque donnée (réponse de type `ITEM`) pour faciliter l'extraction de celles-ci (avec `sscanf()` par exemple) au cas où il y a la présence d'espace dans ces données (dans le nom ou la catégorie par exemple).

Exemple sur une interface en ligne de commande :

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier Édition Affichage Rechercher Terminal Aide
Que souhaitez-vous faire?
    1 - Consulter mon historique de vente
    2 - Gérer mes ventes en cours
    3 - Effectuer une nouvelle vente
    0 - Quitter le Programme
Choix[1|2|3|0]: 2
1 -      UID:1300285939
      Nom: Test 1
      Catégorie: test
      Prix de départ: 160.000000 Euro
      Dernier enchère: 0.000000 Euro
      Quantité: 20

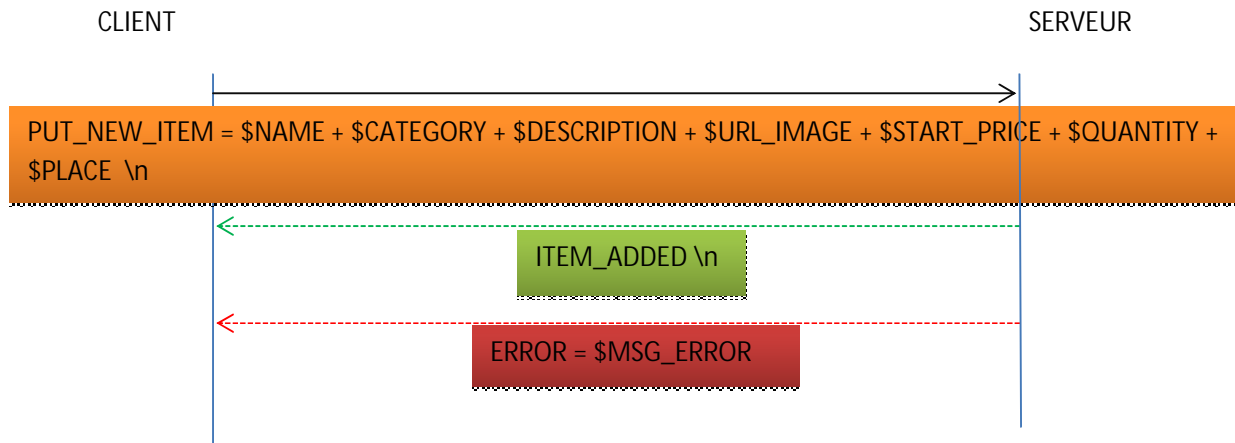
FIN D'AFFICHAGE
Numéro d'item [1 - 1]: 1
Que voulez-vous faire ?
1 - Supprimer l'enchère
Permet de supprimer une enchère en cours.
L'objet sera supprimé du serveur.

2 - Annuler l'enchère
Permet d'annuler une enchère.
Tous les compteurs de l'objet seront remis à zéro.

3 - Terminé une vente
Permet de mettre fin à une vente.
Le dernier à avoir enchérit l'objet sera l'acheteur de cet l'objet.

0 - Quitter cette fonctionnalité
Choix[1|2|3|0]: 1
L'enchère a bien été supprimé
Que souhaitez-vous faire?
    1 - Consulter mon historique de vente
    2 - Gérer mes ventes en cours
    3 - Effectuer une nouvelle vente
    0 - Quitter le Programme
Choix[1|2|3|0]:
```

3. Ajout d'un nouvel item



Exemple sur une interface en ligne de commande :

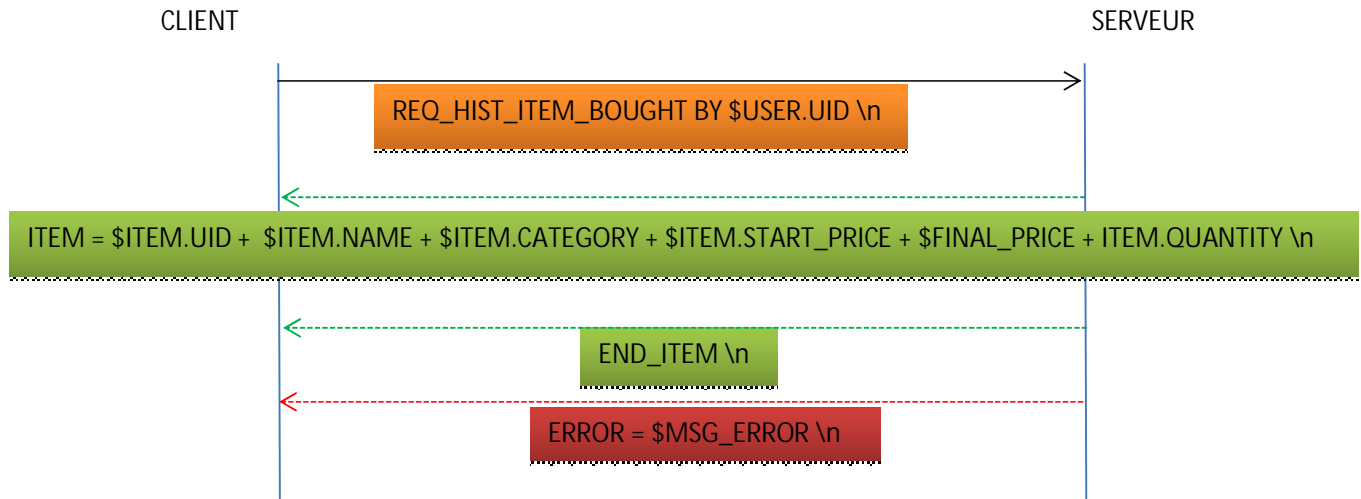
```
manavai@debian: /mnt/hgfs/cbay/client
Fichier  Édition  Affichage  Recherche  Terminal  Aide

Tentative de connexion 1 ...
ConnectÃ© en tant que: test
Choix du mode:
    1 - Mode Acheteur
    2 - Mode Vendeur
    3 - Mode Administrateur
Choix[1|2|3]: 2
Que souhaitez-vous faire?
    1 - Consulter mon historique de vente
    2 - Gérer mes ventes en cours
    3 - Effectuer une nouvelle vente
    0 - Quitter le Programme
Choix[1|2|3|0]: 3
Nom: Test 1
Catégorie: test
Description: ceci est un test
URL d'une image: http://image.com/test
Prix (tous le lot): 160
Quantité: 20
Adresse de l'objet: rue du test 31300 Toutest
L'enchère a bien été ajouté
```

V. Mode Acheteur

Le mode acheteur permet de mettre des enchères sur des objets après les avoir consultés dans une liste de catégorie. Par ailleurs l'Acheteur a la fonctionnalité de pouvoir regarder ses historiques d'achats, de gérer ses enchères, et d'y effectuer des opérations (augmenter une mise, annuler une mise, etc.). Enfin il pourra regarder son historique d'achat.

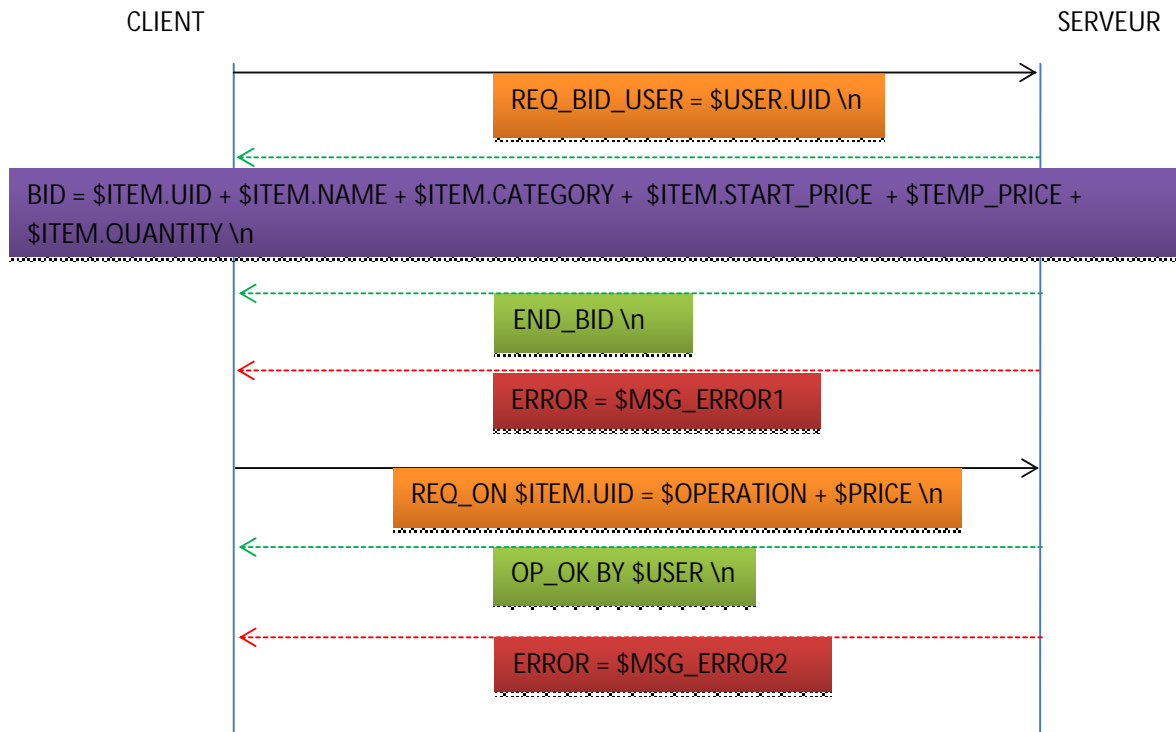
1. Affichage historique des achats



Lorsqu'un utilisateur désire consulter son historique d'achat il envoie une requête à l'application serveur avec son identifiant unique. Ensuite un algorithme de recherche effectue une recherche sur les objets en faisant la correspondance entre l'UID de l'acheteur final d'un objet et l'UID de l'utilisateur. Au fur et à mesure de la correspondance, l'application serveur envoie au client les caractéristiques ci-dessus d'un objet. Lorsqu'il n'y a plus de correspondance, le serveur envoie au client l'information « END_ITEM » pour signifier qu'il n'y a plus d'objet à afficher.

L'affichage des objets est à la charge du développeur de l'application cliente.

2. Gestion enchères en cours



Le gestionnaire des enchères en cours permet d' :

- Afficher les objets en cours d'enchère où vous avez effectué une mise.
- Augmenter une mise sur un objet ;
- Annuler une enchère.

Lorsque l'utilisateur envoie une requête avec son UID d'utilisateur, le serveur effectue un algorithme de recherche sur les objets en se basant sur un fichier de log (ou de journalisation) qui répertorie toutes les opérations qui se sont passées sur le serveur. Ainsi, l'algorithme fait une correspondance entre l'utilisateur et l'objet. A chaque correspondance, le serveur envoie les caractéristiques de l'objet et quand il n'y a plus de correspondance, le serveur envoie l'information « END_BID ». La gestion de l'affichage est à la charge du développeur de l'application cliente. De plus seules les enchères encore en cours seront affichées. Les enchères terminées ne seront pas consultables par l'utilisateur s'il n'est pas l'acheteur final.

Ensuite l'utilisateur peut sélectionner un objet et y effectuer les opérations citées ci-dessous :

\$OPERATION :

UP_BID : Permet d'augmenter une enchère. Cette opération est doit être suivi d'un prix supérieur à la mise en cours.

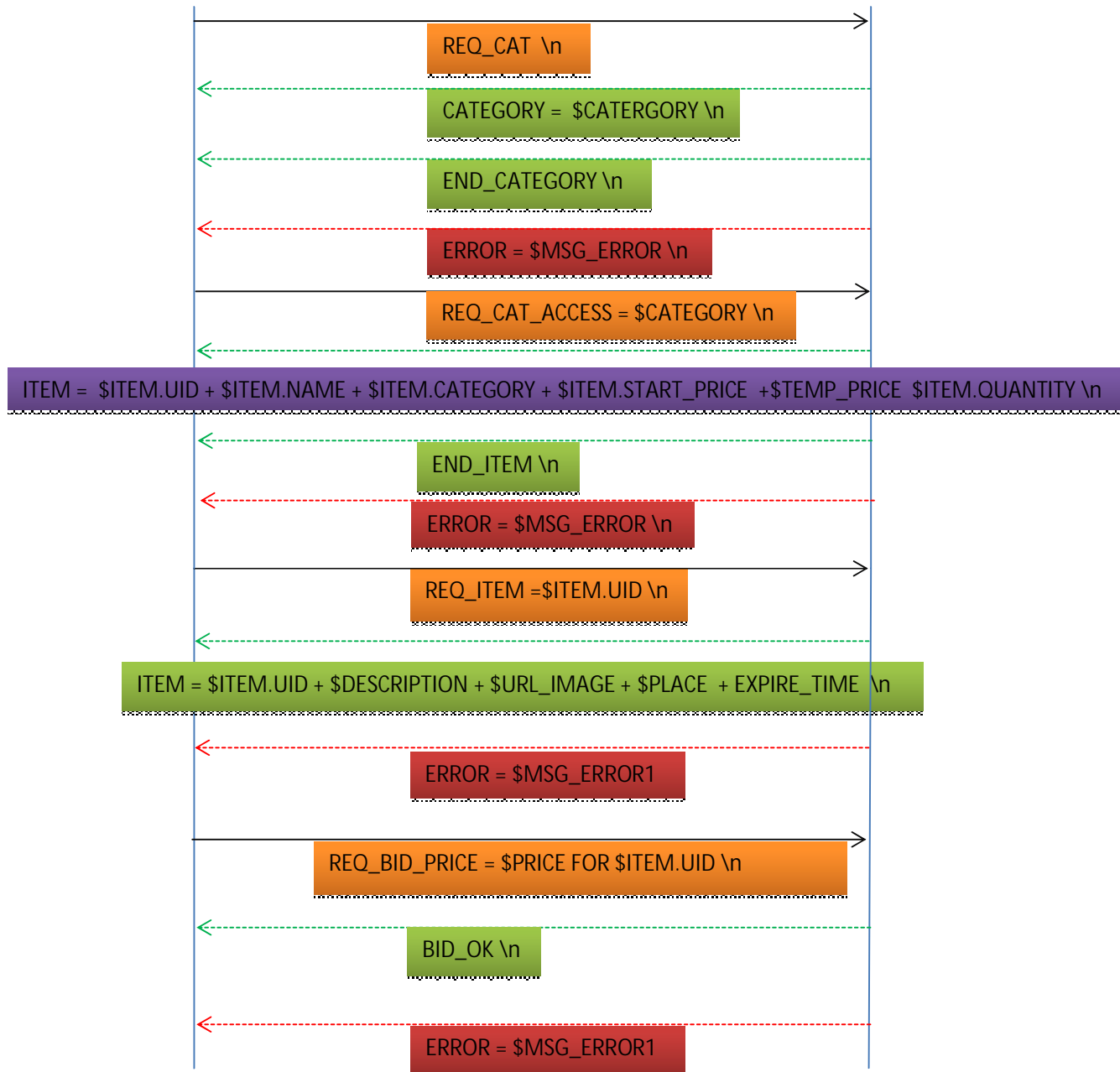
CANCEL_BID : Permet d'annuler une enchère. Le prix inclus dans la requête doit être nul ou négatif.

Modification : Des séparateur « + » ont été ajouté entre chaque donnée (réponse de type BID) pour faciliter l'extraction de celles-ci (avec `sscanf()` par exemple) au cas où il y a la présence d'espace dans ces données (dans le nom ou la catégorie par exemple).

3. Enchère sur un nouvel item

CLIENT

SERVEUR



Dans ce processus d'enchère sur un nouvel objet, on peut distinguer plusieurs étapes. Une première étape consistant à afficher les catégories disponibles sur l'application serveur. Cette première étape n'est pas obligatoire lorsqu'on a défini des catégories sur les deux applications

serveur et cliente. Il s'agit donc ici d'une option offerte au cas où les développeurs voudraient offrir plus de choix dans la sélection des objets.

Les catégories seront envoyées une par une jusqu'à la réception de l'information `END_CATEGORY` ».

Ensuite, une fois la catégorie choisie par la requête `REQ_CAT_ACCESS` », la liste des objets situés dans cette catégories seront envoyé une par une jusqu'à la réception de l'information `END_ITEM` ».

Une fois l'objet sélectionné par la requête `REQ_ITEM` » en fournissant l'UID de celui-ci, Toutes les informations concernant l'objet seront envoyés. L'affichage est à la charge du développeur de l'application cliente.

Enfin, l'utilisateur pourra s'il le désire effectuer une mise sur l'objet. Cette mise doit être supérieure ou égale au prix de départ s'il s'agit de la première mise. Sinon, la mise doit être supérieure à la mise en cours de l'objet. Il est à la charge de l'application d'effectuer des vérifications préalables lors d'une mise, sinon le serveur enverra un message d'erreur.

Modification : Des séparateur `« + »` ont été ajouté entre chaque donnée (réponse de type `BID`) pour faciliter l'extraction de celles-ci (avec `sscanf ()` par exemple) au cas où il y a la présence d'espace dans ces données (dans le nom ou la catégorie par exemple).

Exemple sur une interface en ligne de commande :

```
manavai@debian: /mnt/hgfs/cbay/client
Fichier Édition Affichage Rechercher Terminal Aide
Choix du mode:
    1 - Mode Acheteur
    2 - Mode Vendeur
    3 - Mode Administrateur
Choix[1|2|3]: 1
Que souhaitez-vous faire?
    1 - Consulter mon historique de d'achat
    2 - Gérer mes enchères en cours
    3 - Effectuer un nouvel enchère
    0 - Quitter le Programme
Choix[1|2|3|0]: 3
1 - test

FIN D'AFFICHAGE
Numéro de catégorie [1 - 1] (0 to leave): 1
1 -      UID:1300286534
      Nom: Test 1
      Catégorie: test
      Prix de départ: 20.000000 Euro
      Dernier enchère: 0.000000 Euro
      Quantité: 15

FIN D'AFFICHAGE
Choisissez l'Objet dont vous souhaitez avoir plus d'information.
Numéro d'item [1 - 2] (0 to leave): 1
      UID:1300286534
      Nom: Test 1
      Catégorie: test
      Description: ceci est un test
      URL IMAGE: http://test.com
      Prix de départ: 20.000000 Euro
      Dernier enchère: 0.000000 Euro
      Quantité: 15
      Adresse du vendeur: rue du test
      Temps avant expiration de l'enchère [en min]: 55

Votre montant doit être supérieur à l'enchère en cours ou au prix initial (s'il
n'y pas d'enchère en cours)
Montant [en Euro]: 25
L'enchère a bien aboutit.
Vous pouvez maintenant voir l'objet dans vos enchère en cours.
Que souhaitez-vous faire?
```

VI. Codes d'erreur

Des erreurs pourront être rajoutées, modifiées ou supprimées lors de la programmation des applications par les développeurs. Les erreurs citées ci-dessous ne sont là que pour inspirer les développeurs.

```
// Code d'erreur lors de l'inscription
```

```
0x000 "Unknown Error"  
0x001 "Login is not free"  
0x002 "Login too long"  
0x003 "Password too long"
```

```
//Code D'erreur lors de la connexion d'un utilisateur
```

```
0x010 "Unknown Error"  
0x011 "Login not found"  
0x012 "Bad Password"  
0x015 "You are actually Banned !"
```

```
// Codes d'erreur lors du choix du mode
```

```
0x020 "Unknown Error"  
0x021 "You can not choose admin mode"  
0x020 "Bad mode: not existing (admin, sell, buy)"
```

```
// Mode Administrateur
```

```
//Erreurs liées au changement de mot de passe d'un utilisateur.
```

```
0x040 "You are not and Admin"  
0x041 "Can not modify password: User not found"  
0x042 "Can not modify password: User is an admin"  
0x043 "Can not modify password: New password is the same"
```

```
//Erreurs liées à la suppression d'un utilisateur.
```

```
0x051 "Can not delete user: User not found"  
0x052 "Can not delete user: User is an Admin"  
0x054 "Can not delete user: You are not an admin"  
0x055 "Can not delete user: Unknown Error"
```

```
// Mode Vendeur
```

```
0x060 "Unknown Error"  
0x061 "Can not load your solded Items"  
0x062 "Can not load your current bid"
```

0x064 "Can not add new item"

// Mode Acheteur

0x070 "Unknown Error"

0x071 "Can not load category..."

0x072 "Can not make a bid..."

0x074 "Cant not load hisroty"

0x075 "Bad price: Lower that current bid"

// Erreur général

0x999 "Internal Server Error"