



Projet Informatique: Enchères Distribuées

But du projet:

Le but est de programmer une application en C en mode client/serveur qui gèrera les enchères entre des acheteurs et des vendeurs comme pour le site ebay.

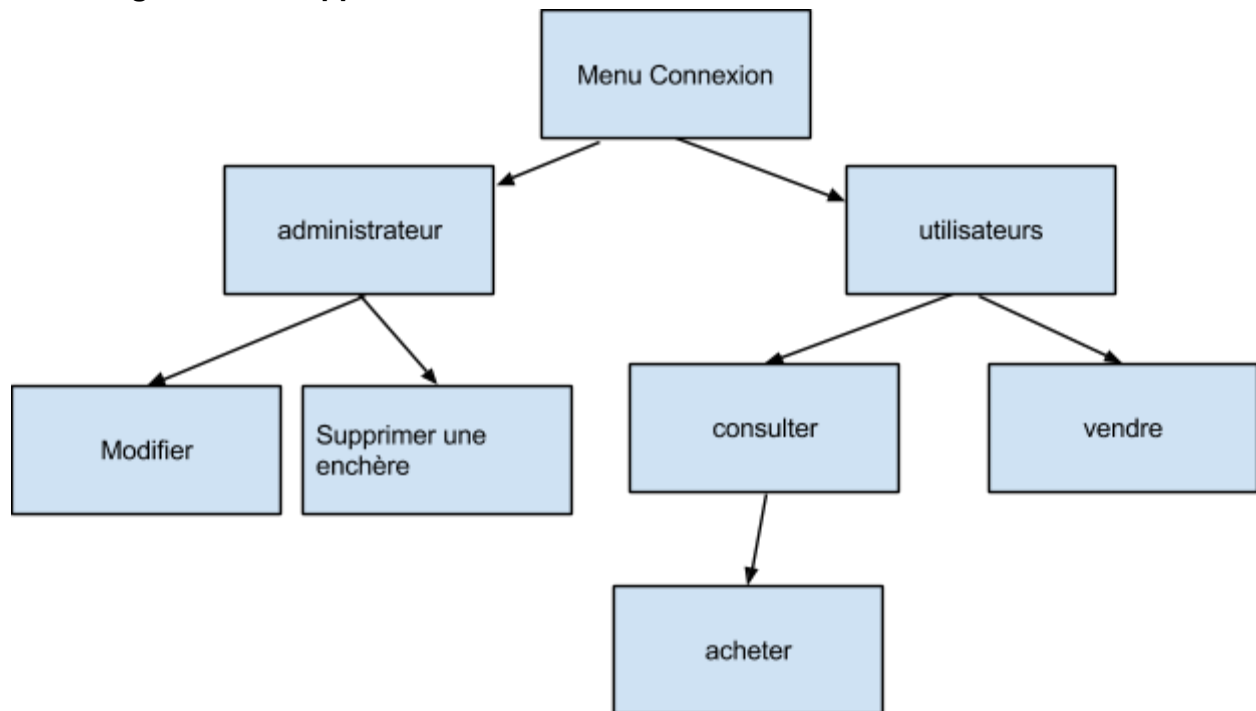
Tout d'abord, un client devra se connecter au serveur pour mettre à disposition des objets à vendre ou tout simplement consulter le catalogue de vente et proposer un prix pour un objet qui l'intéresse.

Un acheteur pourra proposer un prix pour un objet mais l'enchère ne sera prise en compte que si le prix est supérieur au prix en cours et si la date de fin n'est pas dépassé. Le dernier acheteur qui propose le prix le plus haut gagne l'enchère.

Le programme permettra à un acheteur de soit proposer un prix unique, soit proposer un prix maximum avec un 'pas', le serveur gèrera l'augmentation de l'enchère pour cet utilisateur jusqu'à cette valeur maximum.

Le serveur identifiera les utilisateurs et les objets par des identifiants uniques, il enregistrera toutes les informations relatives aux objets et utilisateurs du site dans des fichiers.

Schéma général de l'application:



Les acteurs:

Les différents acteurs devront se connecter au site en utilisant un identifiant et un mode passe défini lors de leur création de compte.

Lors de la connexion, un acteur pourra être aussi bien acheteur ou vendeur. Les administrateurs auront un compte à part pour éviter tout mélange avec des comptes clients.

Les informations:

Les informations sur les objets en cours de vente seront enregistrées dans un fichier "objets".

Les informations sur les utilisateurs seront enregistrées dans un fichier "users".

Les informations sur les enchères en cours seront enregistrées dans un fichier "enchères".

Les informations sur les objets qui ont été vendues seront enregistrées dans le fichier "historique".

Les informations sur les administrateurs seront enregistrées dans un fichier "admin".

Les fichiers:

le symbole \$ permet de séparer différents sujets dans les fichiers.

le symbole # permet de séparer les différentes informations concernant un même sujet.

Le fichier “objet en vente”:

\$identifiant Produit#identifiant utilisateur#Nom objet#date de mise en vente#prix de base#catégorie#description#URL d'une image#quantité#ville de l'utilisateur

le fichier “historique objet”:

\$identifiant Produit#I#Nom objet#date de mise en vente#date de fin de vente#prix de base#prix de vente#catégorie#description#URL d'une image#quantité#ville de l'utilisateur

le fichier “enchère”:

\$Identifiant user#identifiant objet#acheteur1#prix#acheteur2#prix#...#date de mise en vente#date de fin de vente

le fichier “user”:

\$identifiant utilisateur#Nom#Prenom#date de naissance#N° de téléphone#adresse mail#N° de rue#nom rue#ville#code postal#mot de passe#nombre d'objet acheté#nombre d'objet vendu

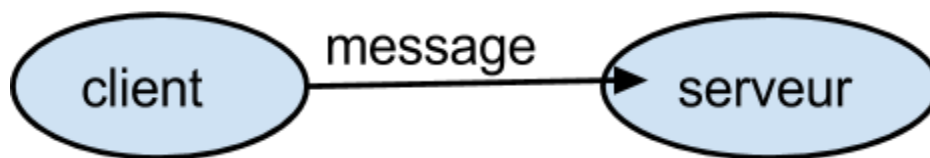
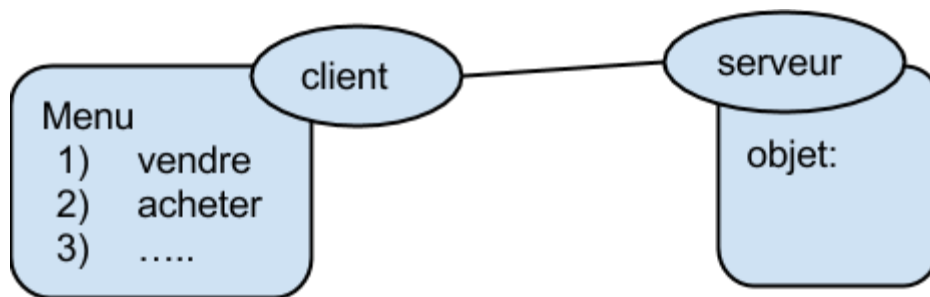
Remarque :

Identifiant utilisateur est un identifiant unique donné par le serveur sous la forme de XXX00 où les deux premiers caractères sont les deux premières lettres du Nom, le 3ème caractère est l'initiale de prénom, deux chiffres qui suivent servent de compteur.

Exemple: Pour l'utilisateur MASSIP Thomas son identifiant sera : MAT01 si il est le premier 'MAT' .

Identifiant Produit est l'identifiant unique attribué par le serveur à un objet sous la forme d'un nombre.

Format des messages:

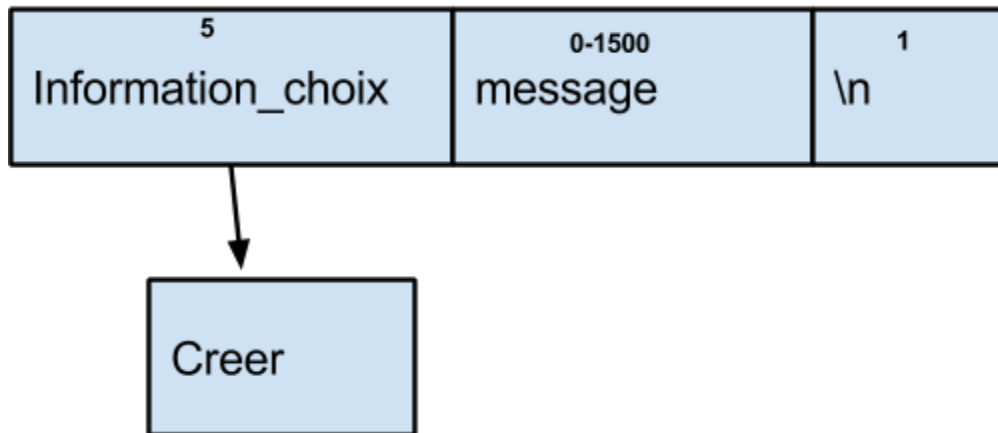


mots clés requêtes	xxxxxx
fin de trame	\n
#	séparateur d'information

Les différents mots clés:

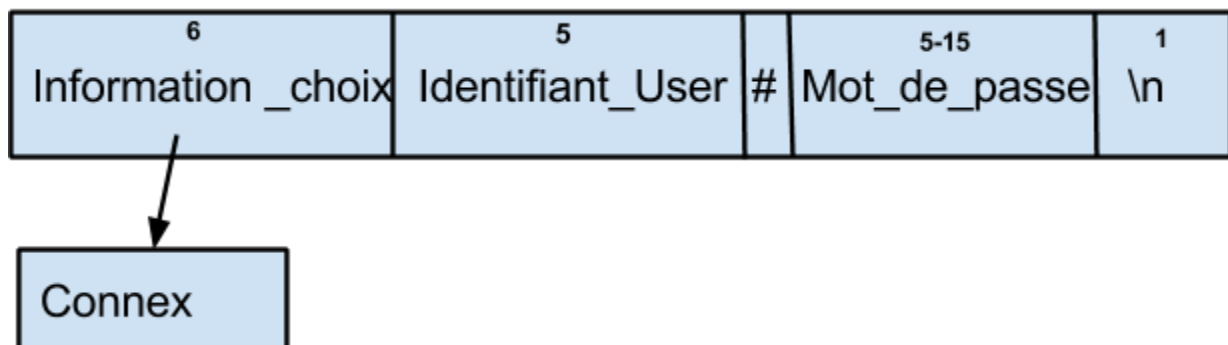
Creer: création de compte
Connex: authentification de l'utilisateur sur le serveur
Consulter: demande à consulter les objets mis en vente
Acheter: proposer un prix pour un objet
Vendre: vendre un objet
Suppr: supprimer un produit de la vente
Deco: information de l'utilisateur de couper la connexion

Message d'inscription compte:

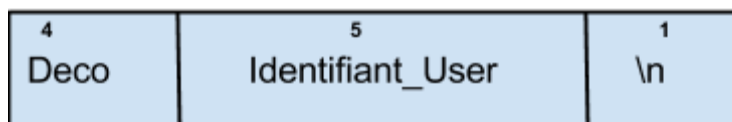


format du "message" = "Nom#Prenom#date de naissance#N° de téléphone#adresse mail#N° de rue#nom rue#ville#code postal#mot de passe"

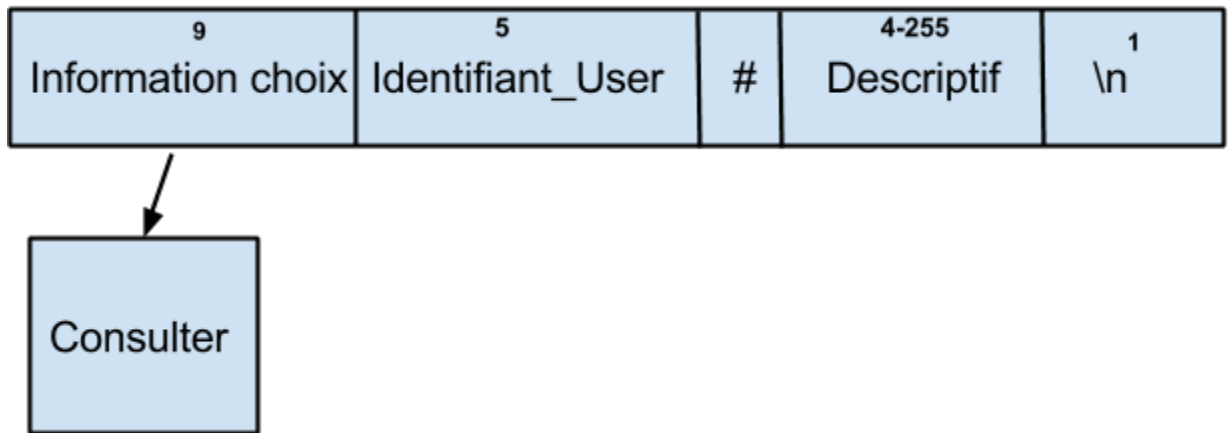
Message "Connexion":



Message "Déconnexion":



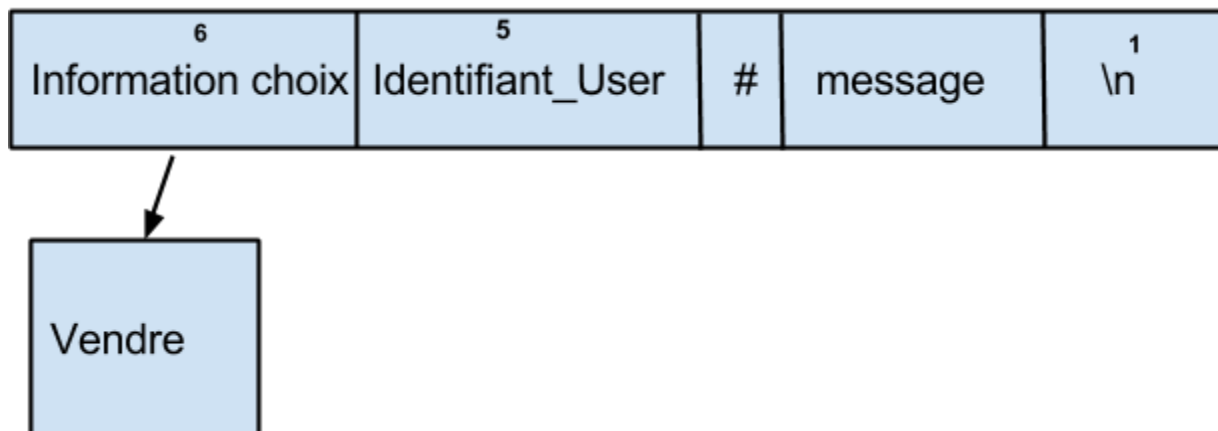
Message consulter:



“Descriptif”:

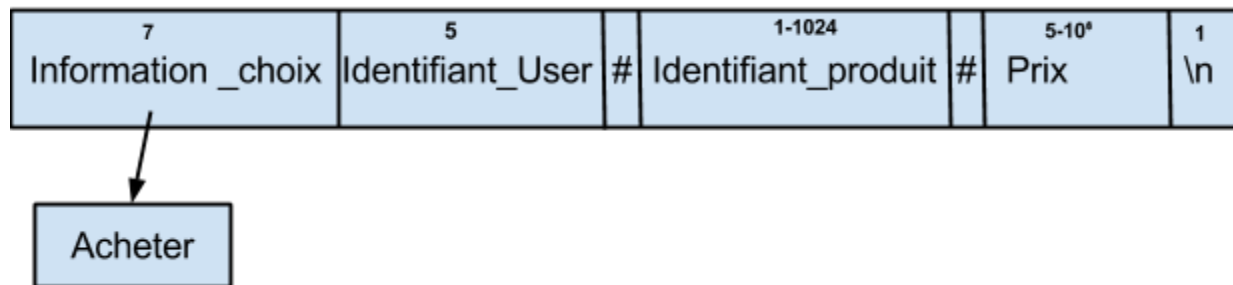
- Si l'utilisateur ne rentre aucun descriptif du produit, le mot clé sera par défaut “Tous”.
- Sinon l'utilisateur pourra entrer un descriptif du produit à rechercher compris entre 4 et 255 caractères. La recherche s'effectuera sur le titre des objets et la description.

Message “Mettre en Vente”:

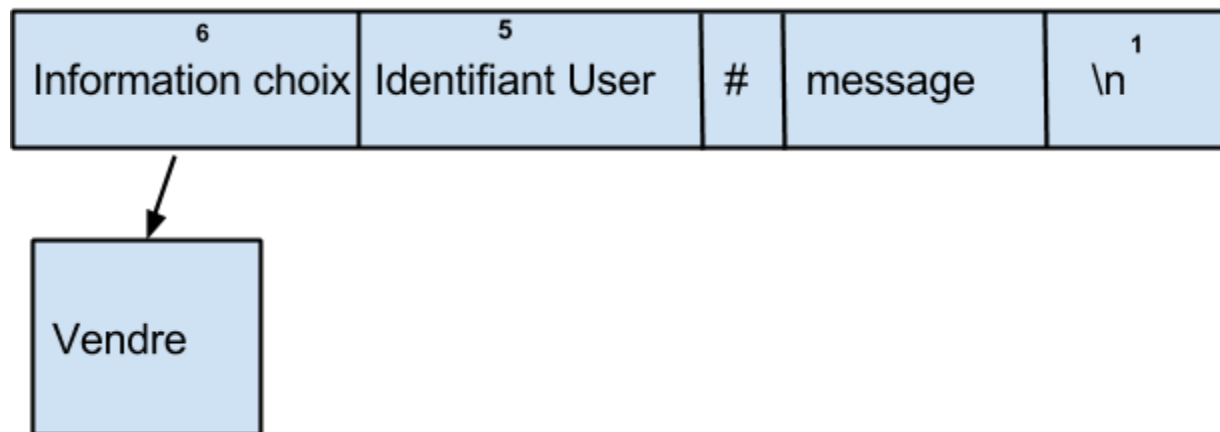


message= “Nom objet#date de mise en vente#date de fin de mise en vente#prix de base#catégorie#description#URL d'une image#quantité#ville où se situe l'objet”

Message "Acheter":

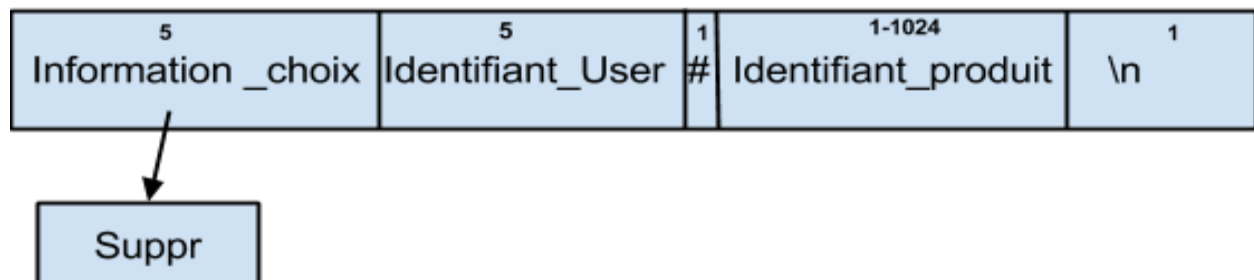


Message "Mettre en Vente":

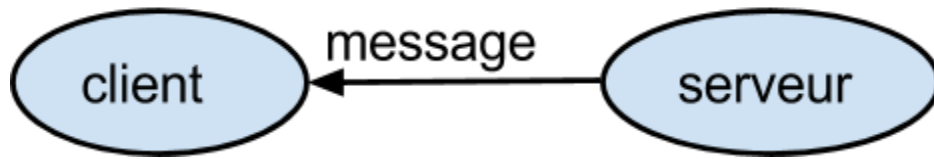


message= "Nom objet#date de mise en vente#date de fin de mise en vente#prix de base#catégorie#description#URL d'une image#quantité#ville où se situe l'objet"

Supprimer une information:



Remarque: Uniquement dans le cas où un utilisateur ne veut plus vendre un produit.



Message “réponse serveur”:

Connexion OK: la connexion est réussie



Erreur connexion: le serveur ou le serveur rencontre un problème



Connexion terminée: le client a terminé



Message Ok: le message est correct



Enchère :

EnchereOK : La date de l'enchère est inférieure à la date de fin d'enchère (12/10/2014 < 21/10/2014) et le prix de l'enchère est supérieur au prix actuel.

⁹ EnchereOK	² \n
---------------------------	--------------------

Message Acheteur: Type de message que l'acheteur reçoit lors d'une enchère.

Message Enchere	² \n
-----------------	--------------------

Message Enchère (pour l'acheteur):

'L'enchère sur \$Nom_Produit est terminé, vous ne l'avez pas remporté.'

'L'enchère sur \$Nom_Produit est terminé, vous l'avez remporté au prix de \$Prix €.'

Message Enchère (pour le vendeur):

'L'enchère sur \$Nom_Produit est terminé, le prix est de \$Prix est l'acheteur est \$Nom_Utilisateur.'

Message Erreur: le message est incorrect

10 MsgError	Type_Erreur	1 \\n
----------------	-------------	----------

Type_Erreurs:

Information_choix : Le mot clé 'Information_choix' n'a pas été reconnue par le serveur

Identifiant_user : L'identifiant est incorrecte

Identifiant_produits : L'identifiant du produit est incorrecte

mot_de_passe : Le mot de passe est incorrecte

Erreur_fichier : fichier ne s'ouvre pas

Erreur_image : image_url ne s'ouvre pas

ErreurEnchere : La date de l'enchère est supérieur a la date de fin d'enchère ou le prix de l'enchère est inférieur au prix actuelle.

Catalogue:

Le catalogue sera un menu ou le client visualise l'objet (prix, URL, description ect..), il aura alors trois possibilités :

Suivant : Permet d'aller à l'objet suivant concernant ça description de recherche.

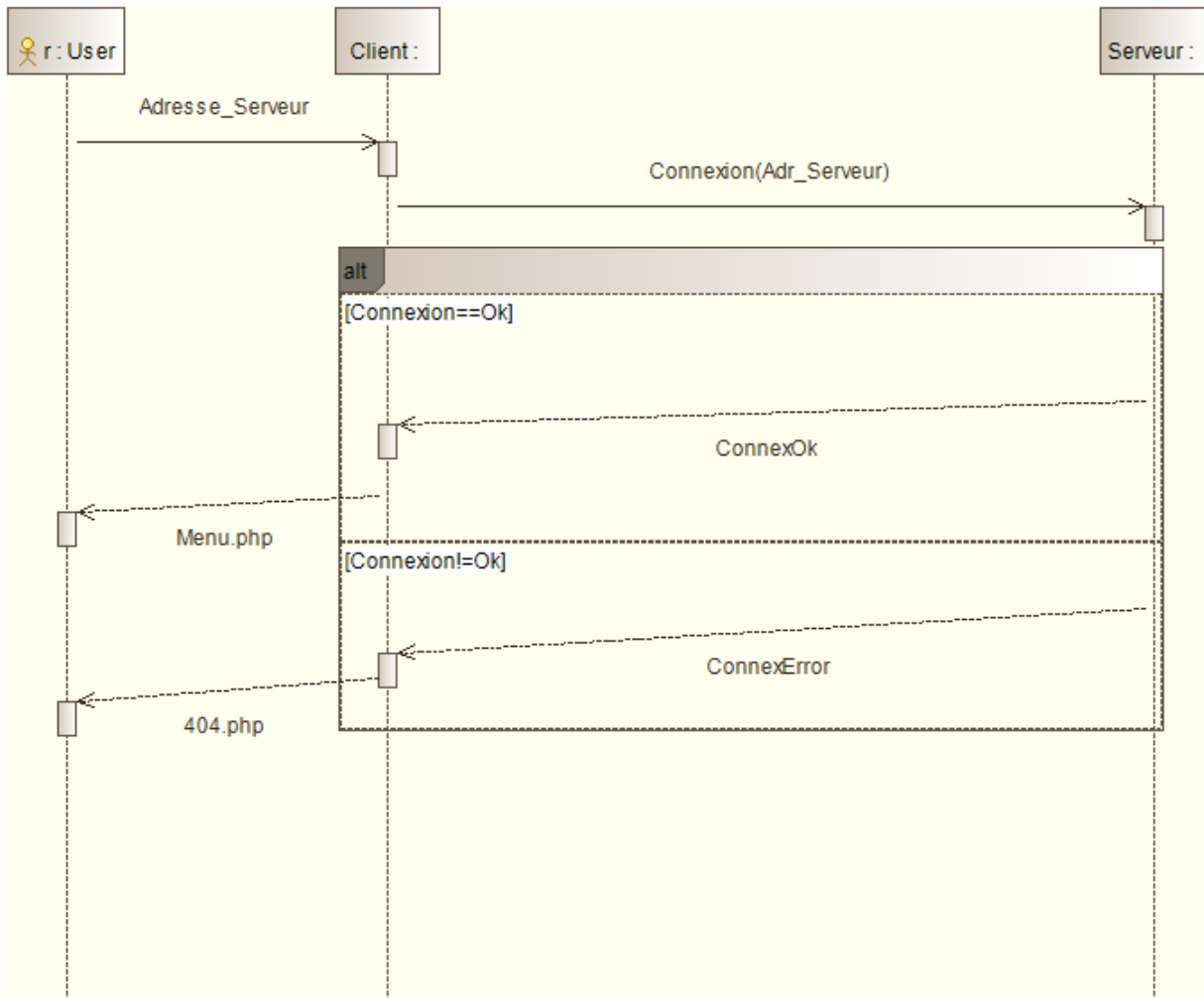
Précédent : Permet de revenir sur l'objet précédent

Acheter : Permet d'enchérir sur l'objet en cours de visualisation.

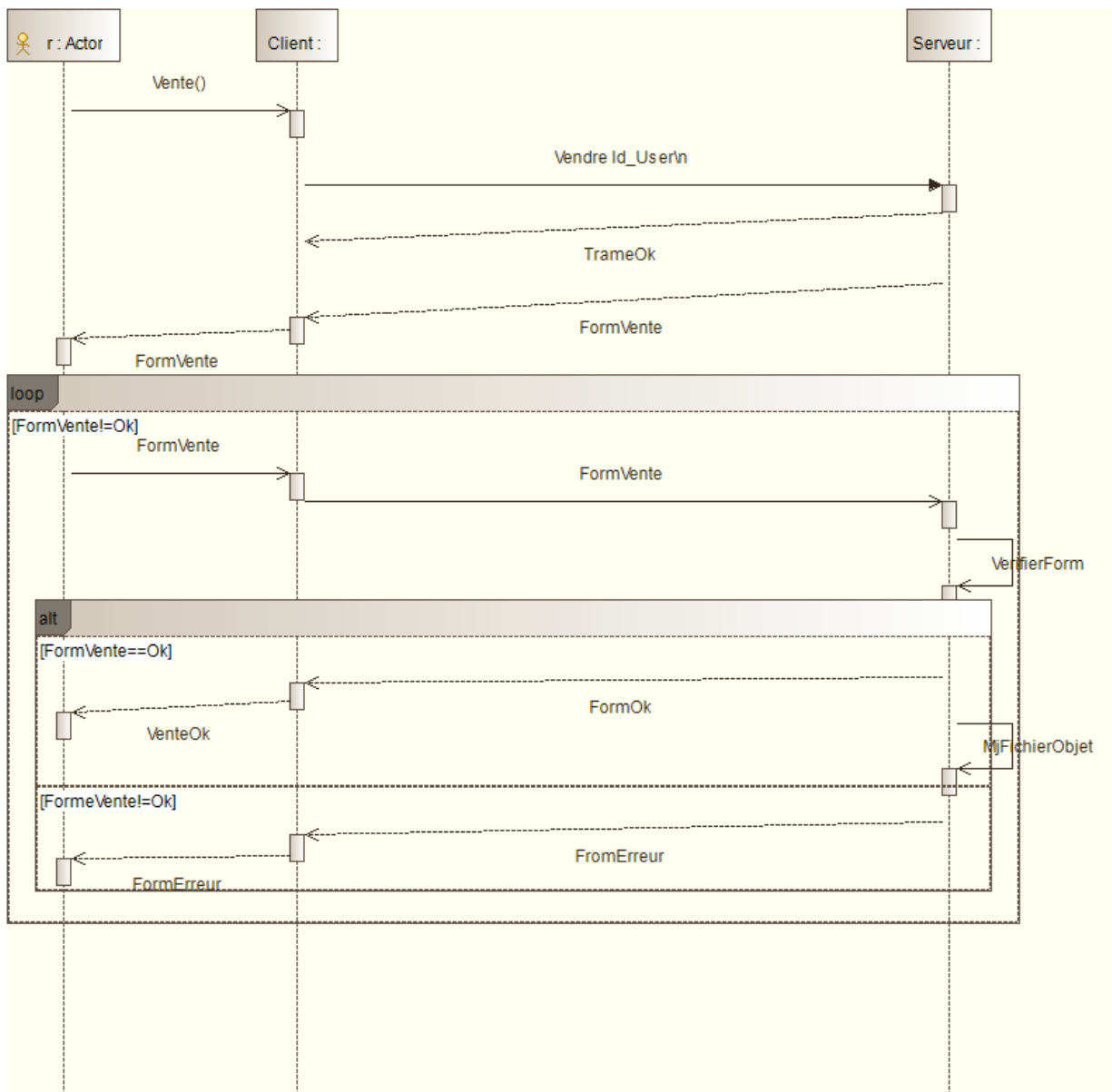
Quitter: Permet de revenir au menu.

Exemples de dialogue:

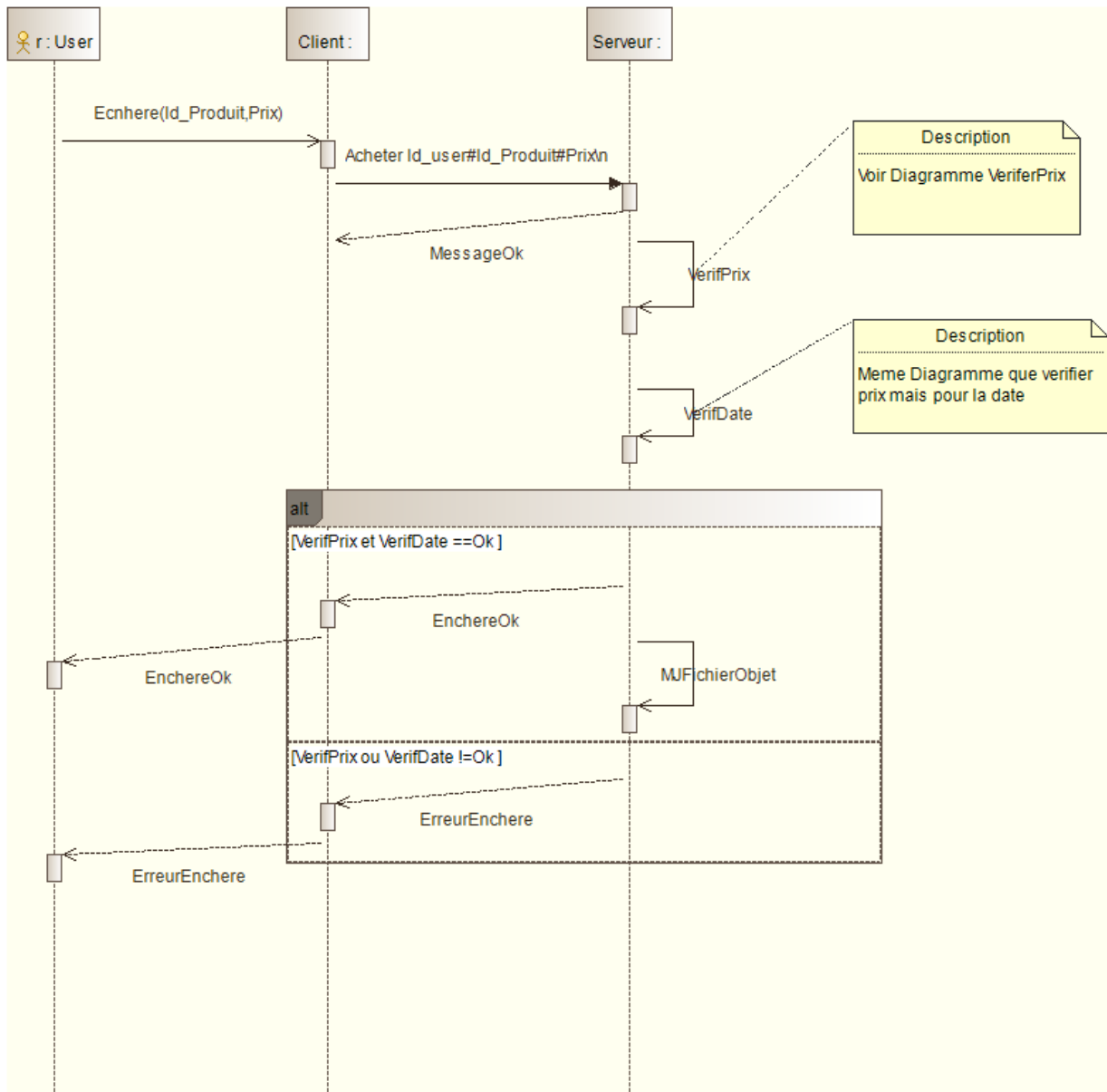
Connexion Client/Serveur:



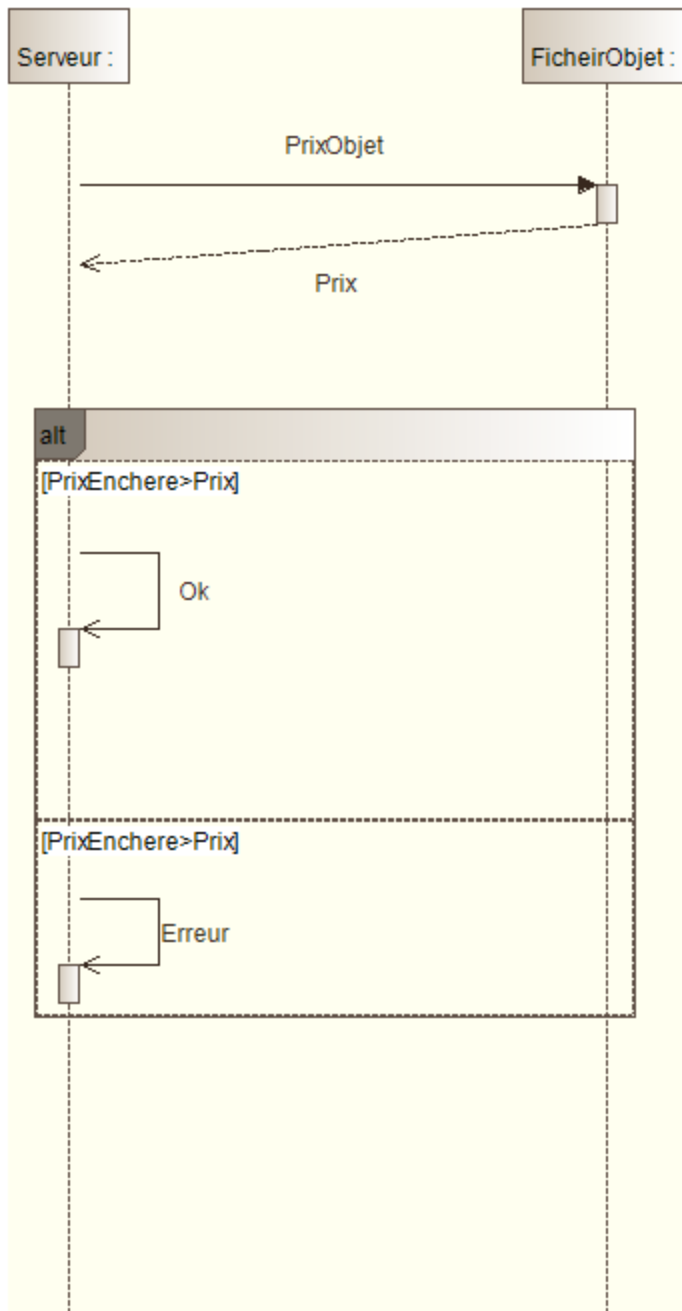
Pour tout les diagrammes suivant nous parton du principe qu'il n'y a pas d'erreur de message
 Mise en vente d'un objet :



Achat d'un objet :



Vérifier le prix d'un objet :



Algorithme:

Algo: mainClient:

Debut

 connexionServeur("adresse serveur");

 si connexionServeurOk alors

 faire

 msgConnexion<-MenuConnex(); //on recupère l'identifiant de connexions et le mode de passe dans une chaîne de caractères

 émission(msgConnexion); // on envoie les informations au serveur

 reception(msg); // msg retour serveur

 si msg="identif OK" alors /*l'utilisateur peut accéder au menu */

 Menu(); /* dans la fonction menu l'utilisateur interagira avec le serveur

 FinSi

 TQ(quitter!=0 && msg!="fin de connexion");

 Sinon afficherEcran("connexion serveur impossible");

 Finsi

FIN algo

Algo: mainServeur

DebutAlgo:

 AttenteClient();

 message = Reception(); //on reçoit le message

 switch(debut(message)){ // on test le premier mot de la chaîne message

 case create: creationDeCompte(message) // on créé un compte client;

 case connect: controleIDUtilisateur(message); // contrôle de l'id et mot de passe de l'utilisateur

 case consulter: consulterProduit(message); // le client demande à consulter les produits du catalogue

 case acheter: achatProduit(message); // le client propose une enchère pour un produit

 case vendre: vendreProduit(message); // le client propose un mise en vente d'un objet

 case quitter: terminaisonClient();//le client demande à se déconnecter du serveur

 case default: MsgError(message); // aucun mot clé trouvé, on envoie un message

d'erreur

 }

FinAlgo