EPREUVE : DEVELOPPEMENT DES APPLICATIONS INFORMATIQUES (DAI)

La société marocaine 'DSIGS' spécialisée dans la vente en ligne veut implanter un programme de gestion de stock afin d'automatiser son système d'information.

La figure suivante représente le modèle logique de données élaboré par un analyste programmeur :

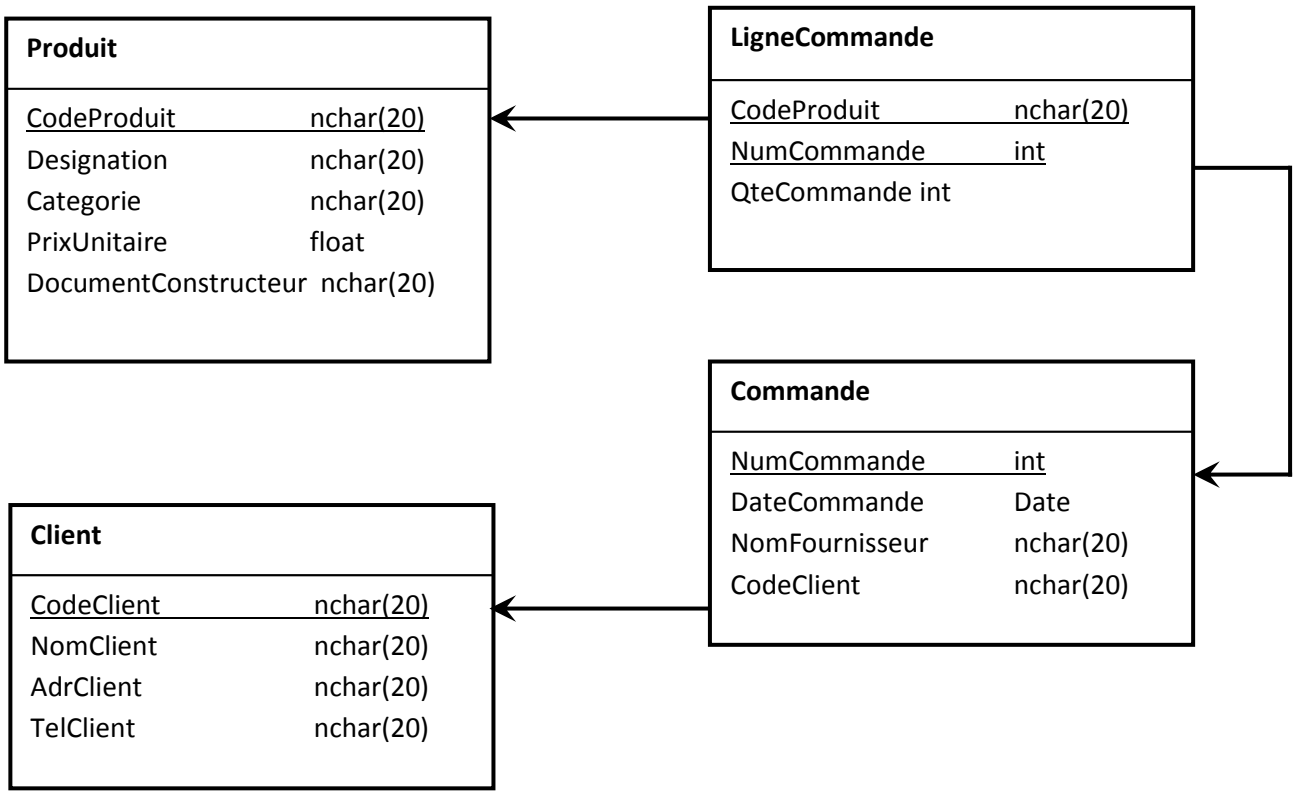

FIGURE 1 : MLD DE L' APPLICATION

PARTIE 1 : PROGRAMMATION ORIENTE OBJET (JAVA) (13pts):

La structure des classes que l'on veut réaliser est la suivante :

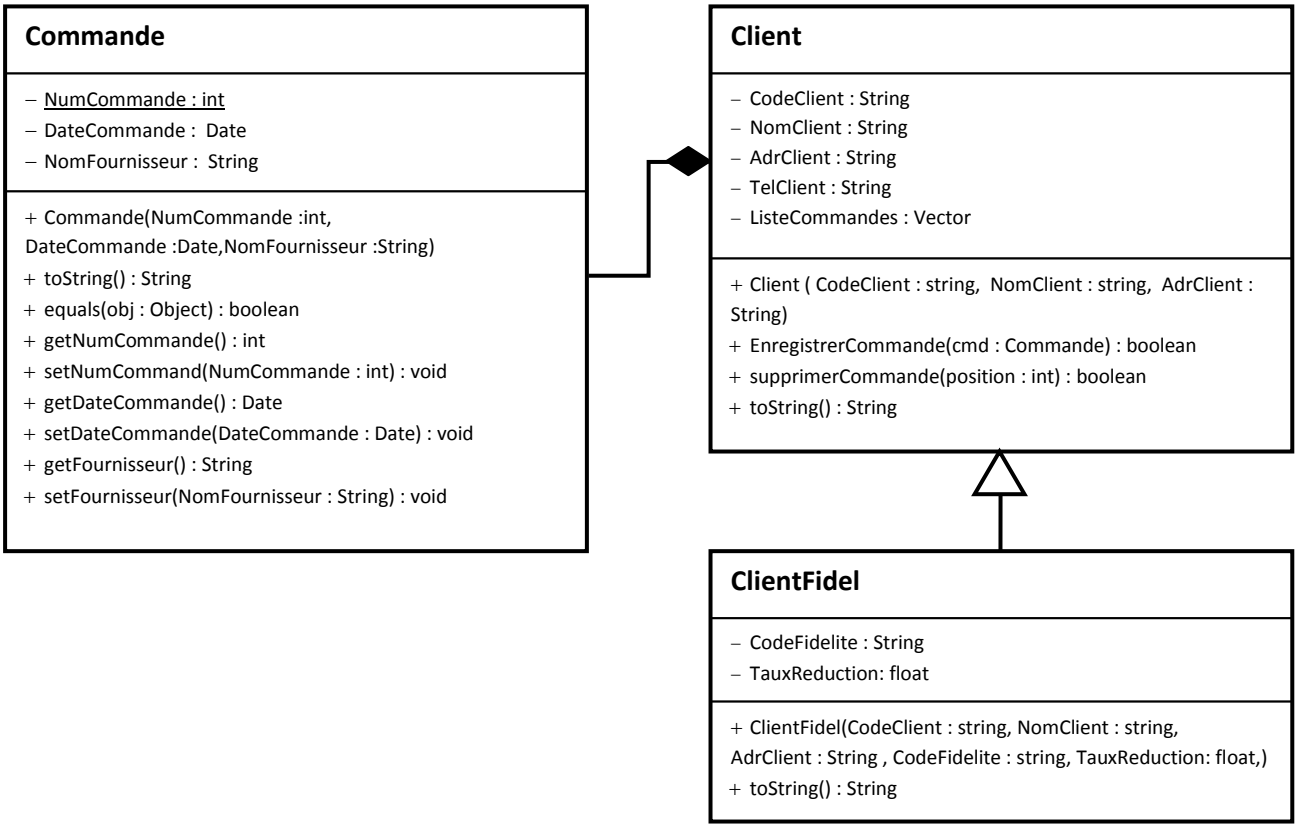

FIGURE 2 : CLASSES DE COMMANDE ET CLIENT

Travail à faire:

1. **Réaliser la classe Commande qui a comme attributs (4pts):**
    - NumCommande de type entier.
    - DateCommande de type Date.
    - NomFournisseur de type String

    Elle a comme méthodes :
    - Un constructeur avec arguments
    - Des accesseurs (getters)
    - Des mutateurs (modificateurs ou setters)
    - La méthode redéfinie toString() retournant une chaine contenant toutes les informations d'une commande.

- La méthode redéfinie **equals(Object obj)** qui compare deux commandes et prend comme critère de comparaison le **NumCommande**.

2. **Quelles sont les modifications à apporter à la classe commande pour faire son *clonage*?(1pt)**
3. **Réaliser la classe Client qui possède comme attributs (4pts):**
    - CodeClient de type String
    - NomClient : nom du client de type String
    - AdrClient : adresse client de type String
    - TelClient: telephone du client de type String
    - **ListCommandes** de type Vector qui contiendra la liste des commandes effectuées par ce client.

    Cette classe contient comme méthodes :
    - Un constructeur d'initialisation qui initialise les trois premiers attributs par les arguments reçus et instancie un nouveau vecteur.
    - La méthode **EnregistrerCommande** permettant d'ajouter une nouvelle commande au vecteur **ListCommandes.**
    - La méthode SupprimerCommande permettant de supprimer une commande du vecteur en se basant sur le NumCommande passé en paramètre, si ce numéro n'existe pas, cette méthode retourne **false** sinon elle retourne **true**.
    - La méthode redéfinie **toString** permettant de retourner toutes les informations d'un client ainsi que la liste de ses commandes, s'il en dispose.
4. **Un client peut être un client fidèle, créer la classe *ClientFidel* (2pts)**

    Cette classe contient comme attribut :
    - CodeFidelite de type String
    - TauxReduction de type float

    Cette classe contient deux méthodes :
    - Un constructeur d'initialisation.
    - La méthode redéfinie **toString**() retourne une chaine contenant toutes les informations concernant une commande.
5. **Donner un programme de test de ces classes (2pts):**
    - Créer deux commandes.
    - Créer deux objets de type Client l'un instancié sur la classe **Client** l'autre sur la classe ***ClientFidel***.

- Ajouter une commande pour chaque client
- Afficher ces objets.

PARTIE2 : DEVELOPPEMENT D'APPLICATION CLIENT/SERVEUR (JAVA)(9pts)

La société DSIGS veut réaliser une application de gestion à distance.

Description de l'application :

L'utilisateur peut accéder à partir de son poste au serveur de base de données et transférer le fichier « document constructeur » de Produit. Pour permettre cet accès, un serveur d'application TCP « Middleware » est installé sur le serveur, dont le rôle est d'ouvrir et envoyer le fichier concernant un produit donné. Le nom de fichier est disponible dans la table Produit au champ DocumentConstructeur. L'identifiant du produit sera émis par l'application Cliente.

Le schéma de ce réseau de transfert de données est le suivant:

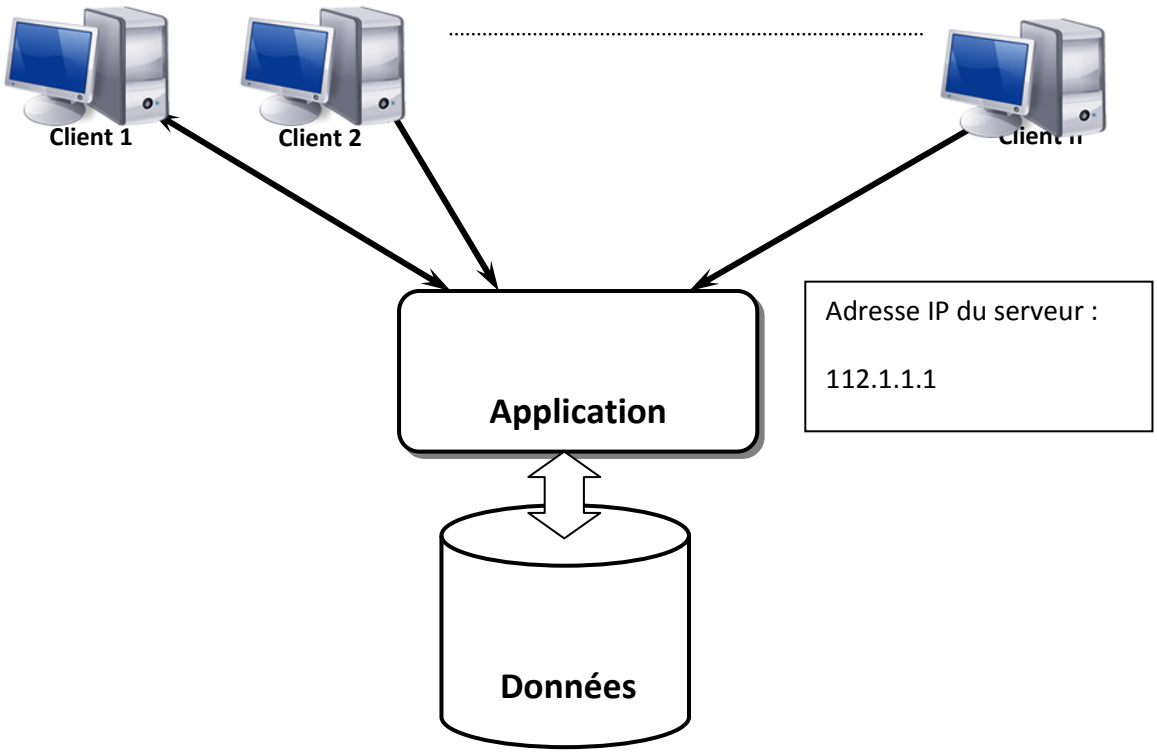

FIGURE 3 : SCHEMA DE RESEAU DE TRANSFERT DE DONNEES

1- Questions de cours (2pts):

a- L'architecture décrite peut être qualifiée de 2-tiers ou de 3-tiers ?

b- Donner le Modèle de Gartner correspondant (selon la question a).

c- Définir une Socket. Quelle est la différence entre socket TCP/IP et UDP ?

d- Définir le middleware et donner des exemples.

2- Le programme coté client (3pts):

Ecrire un programme client en respectant les étapes suivantes:

- Demande de connexion au serveur.
- Saisir au clavier et envoyer le nom du fichier au serveur.
- Recevoir les données envoyées par le serveur (contenu du fichier ligne par ligne (String)) et les enregistrer dans une structure de données (tableau, Vector, ArrayList…)
- Fermeture de la connexion à la fin (réception d'une ligne vide 'NULL').

3- Programme cote serveur (3pts):

Supposant que le serveur traite un seul client à la fois donner son programme en respectant les étapes suivantes:

- Acceptation de la connexion
- Recevoir le nom de fichier demandé par le client et envoyer son contenu (ligne par ligne) (prévoir une gestion d'erreur dans le cas de fichier inexistant).
- Fermeture de la connexion.

4- Si on veut que le serveur traite simultanément plusieurs clients à la fois, qu'elle est la solution convenable (on ne demande pas le code) ? (**1pt**)

PARTIE 3 : ENVIRONNEMENT DE DEVELOPPEMENT INTEGRE (VB.NET) (10pts).

Gestion des lignes de commandes :

L'utilisateur de la société DSIGS dispose d'une interface sous VB.NET permettant de gérer les lignes de commandes, cette gestion consiste à :

- Ajouter une ligne de commande.
- Supprimer une ligne de commande.
- Rechercher une ligne de commande.

L'IHM est illustrée par la figure suivante :

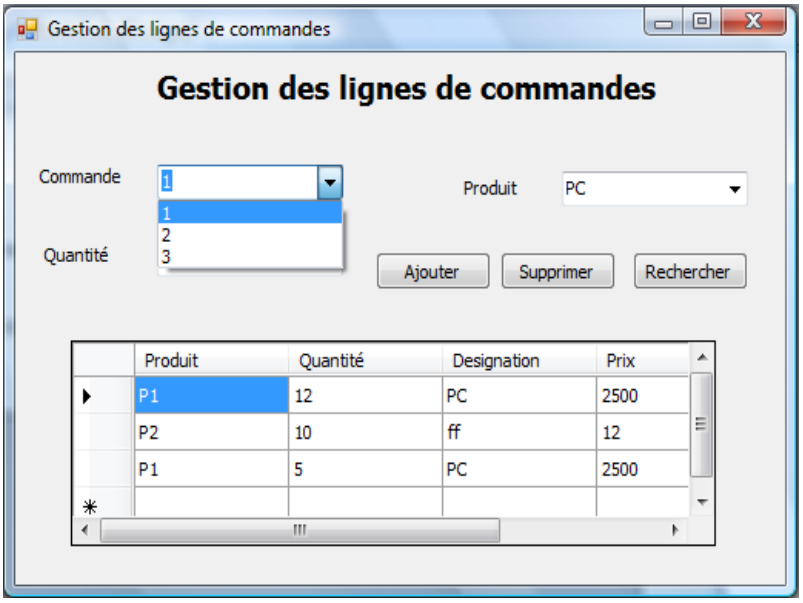

Figure 5 : IHM de gestion des lignes des commandes

Notre base de données intitulée '**magasin'** est de type '**SqlServer'** (nom du serveur :'**localhost'**)

Les attributs de chaque élément du formulaire sont donnés dans le tableau suivant :

Elément du formulaire	**Attribut**
Zone de texte	txtQuantite
comboBox	cmbCommandes
comboBox	cmbProduit
Bouton	Ajouter,Supprimer,Rechercher
DataGridView	DataGridView

Travail à faire :

1- Donner le code de la procédure **Connecter** permettant de se connecter à la base de données (**1.5pts**).

2- Donner le code qui va permettre de Charger le **Dataset** par les trois tables : **Commande**, **Produit** et **LigneCommande**(**1.5pts**).

3- Donner le code de la procédure **remplirCommandes** permettant de remplir le comboBox **cmbCommandes** par la liste des numéros des commandes (**1pts**).

4- Ecrire les instructions de la procédure **Afficher** permettant de remplir l'objet DataGridView constitué par la jointure de trois tables dont les champs sélectionnées sont : produit(CodeProduit), Quantité(QteCommande), Designation et prix unitaire (**2pts**).

5- Programmer le bouton **Ajouter** permettant d'ajouter une nouvelle ligne de commande (**2pts**).

6- Donner le code du bouton **Rechercher** permettant de lister tous les commandes qui ont un numéro de commande lu par une boite de commande (inputBox) (**2pts**).

PARTIE4 : DEVELOPPEMENT WEB(8pts).

On veut faciliter l'enregistrement des clients de la société DSIGS via une application web.

L'application va permettre aux utilisateurs d'effectuer des saisies de données dans un formulaire (figure 6) et de les enregistrer dans la base de données '**magasin**' qui contient la table ***Client (id client, nom, prenom, age, adresse, ville, mail).***

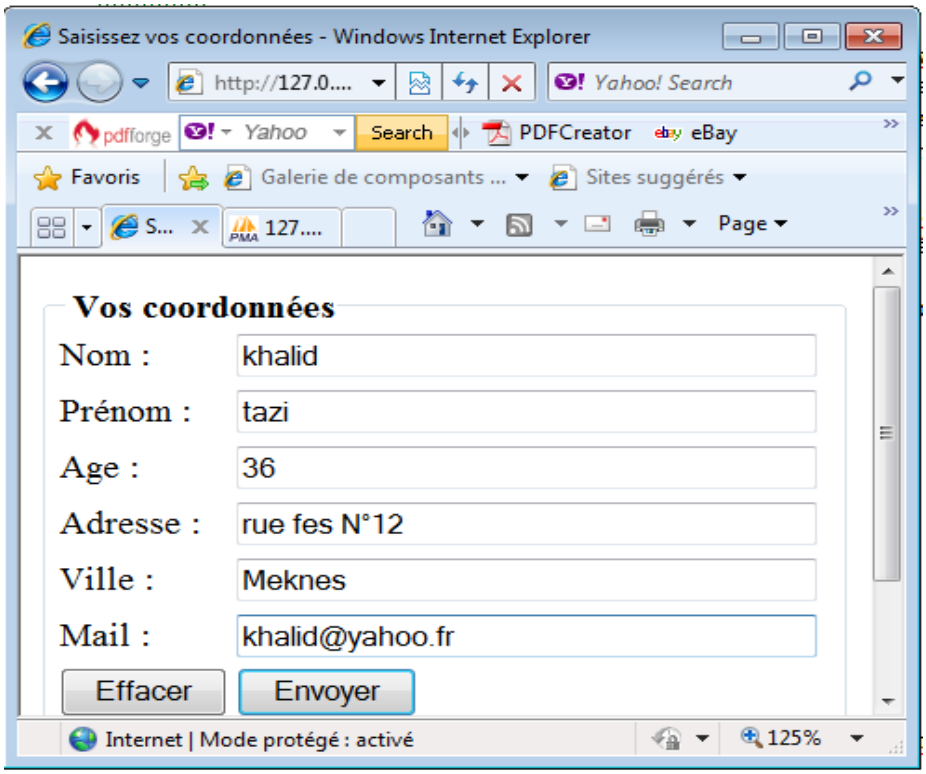

FIGURE6 : FORMULAIRE D'INSERTION DES COORDONNEES

Après le remplissage de tous les champs du formulaire, un clic sur le bouton « Envoyer » enregistre les coordonnées dans la table et une boite de dialogue s'affiche avec le message suivant : « ***vous êtes enregistré votre numéro de client est :9*** », avec « 9 » représente le **id_client** récupéré (exemple on peut utiliser la fonction **int mysql_insert_id()).** La figure ci-dessous représente cette situation. Sinon le script affiche un message d'erreur.

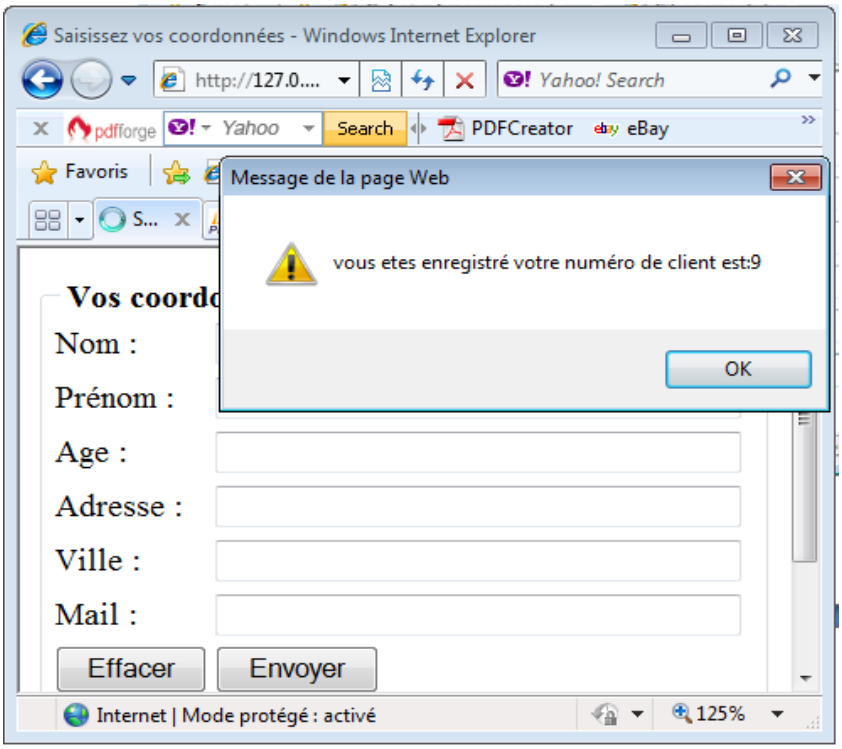

Figure 7 : Message de confirmation d'enregistrement de données

Lorsqu'on clique sur le bouton « OK » de la boite de dialogue le script reste dans la page d'insertion des coordonnées.

Travail à faire :

1. Ecrire le code du formulaire d'insertion (**1pt**).
2. Avec un code JavaScript vérifier que : (**2pts**)
   - tous les champs sont remplis.
   - L'email doit contenir le caractère '@ '.
3. Ecrire les instructions qui permettent de se connecter au serveur et de sélectionner la base de données magasin (1**pt**).
4. Ecrire le script PHP qui permet d'insérer les données saisies dans la table Client (vérifier le résultat de la requête en affichant un message de confirmation d'enregistrement ou un message d'erreur). (**2pts**)
5. On veut garder une trace de nos clients, Ecrire le script PHP qui permet d'insérer les champs du client dans un fichier texte. (**2pts**)