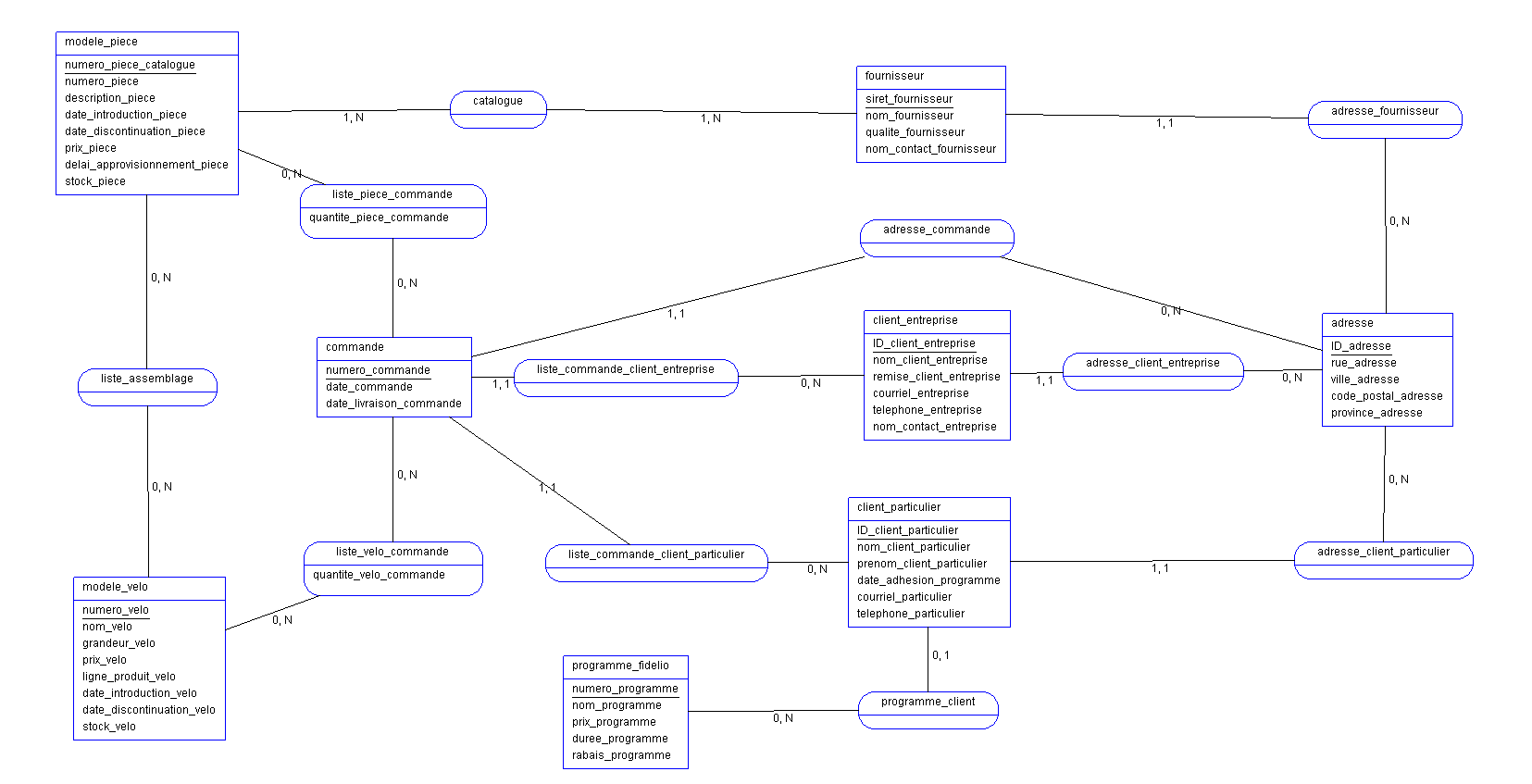
**Rapport projet BDD VeloMax**

1. **Diagramme Entité-Associations**

L’objectif de ce projet est de créer une base de données afin de mieux gérer les ventes et les clients de l’entreprise VeloMax. Le projet englobe aussi l’interface utilisateur sur C# qui doit être le plus ergonomique et intuitif possible.

La première étape est la création du Diagramme Entité/Association qui décrit le squellette de notre base de donnée en fonction des contraintes imposées. J’ai réalisé le diagramme à l’aide du logiciel Analyse SI. (Fichier disponible dans les documents annexes)



Precision sur le diagramme :

\* J’ai décidé de créer une nouvelle table adresse pour différencier l’adresse du client et l’adresse de livraison de la commande effectué par le client.

\*Le catalogue Possède une cardinalité 1,N 1,N ou chaque clé etrangère numero\_piece\_catalogue est associé à une cle etrangere siret\_fournisseur idem pour la liste d’assemblage.

\*Il aurait été possible de créer des tables de commande fournisseur que je n’ai pas souhaité affiché ici pour ne pas complexifié trop le shéma.

1. **Création des tables et peuplement**

Toute les étapes de création des tables et peuplement ont été réalisé sur MySql. Certains peuplement de tables n’étant pas imposé dans les contraintes j’ai inventé des clients des fournisseurs, des commandes ect…

(Fichier disponible dans les documents annexes)

1. **Composition de l’interface WPF c#**

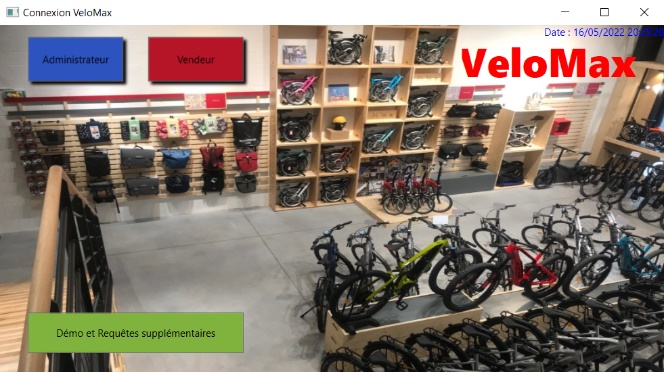
J’ai décidé d’utiliser une interface WPF, ce qui est beaucoup plus intreractif pour l’utilisateur et plus maniable. On se rapproche plus de ce qui pourrait se faire dans la réalité avec ce type d’interface. Chaque fenêtre et composé d’un code XAML pour la disposition des éléments dans la fenêtre ainsi que de classes qui font le lien entre la base de données VeloMax, C# et WPF. Voici la composition de chaque fenêtre :

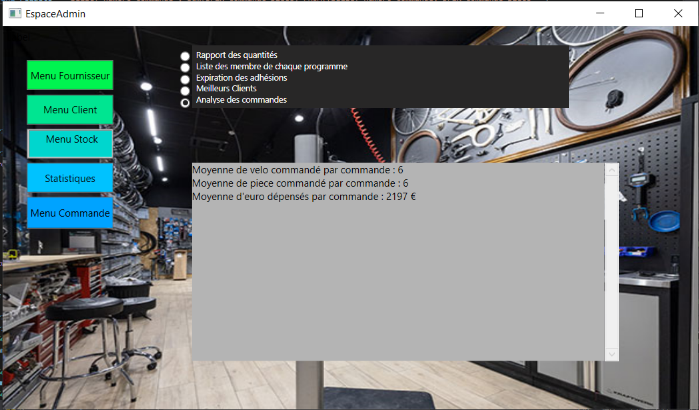
**Fenêtre MainWindow :** Au niveau de la Première fenêtre on retrouve le bouton Démo qui permet une petite démonstration de certaines requêtes. Nous avons aussi ajouter des requêtes supplémentaire (requête synchronisé, auto-jointure et union)

La démo comporte aussi des exportations en JSON et XML accessible dans le fichier bin => debug

On trouve aussi les différents profil de connexion « Administrateur » ou « Vendeur ». En fondtion du profil choisi l’utilisateur n’aura pas accès à la même interface et donc pas accès au même droits. L’administrateur (id,password) = (root/root) (sur l’interface) peut réaliser toutes les requêtes de base (uniquement sur la BDD VeloMax) tandis que le vendeur (bozo/bozo), lui, peut seulment Visualiser certaines donnés (requête SELECT). Sur MySql l’administrateur à comme identifiant bozo et mot de passe bozo et le vendeur bozo2 bozo2 Par manque de temps, l’interface vendeur n’a pas totalement été codé mais un belle aperçu est diponible.

Lors de la connexion le profil de connexion est envoyé dans une variable globale qui sera utilisé par la suite pour l’ensemble des requêtes.





**Fenêtre EspaceAdmin / EspaceVendeur :** Ces fenêtres centralisent tous les menus accessible à l’utilisateur c’est à partir de cette fenêtre qu’il va pouvoir cliquer sur les différentes action qu’il veut effectuer. On peut notamment visualiser les données les diffèrentes requêtes du module statistique

**Fenêtre Menu\_commande :** Dans le Menu des commandes il est possible de faire une nouvelle commande client ou une nouvelle commande Fournisseur. Il est possible de visualiser ces commande mais impossible de les modifier ni de les supprimer, voici pourquoi : Mon projet propose une innovation concernant le statut de la commande, (En attente ,En livraison, Livré ). Je me suis inspiré de la réalité où selon moi les commandes en livraison et livré ne sont plus modifiable car le colis et déjà parti. De plus, on ne peut pas supprimer ces commandes car elle constitue un historique important.

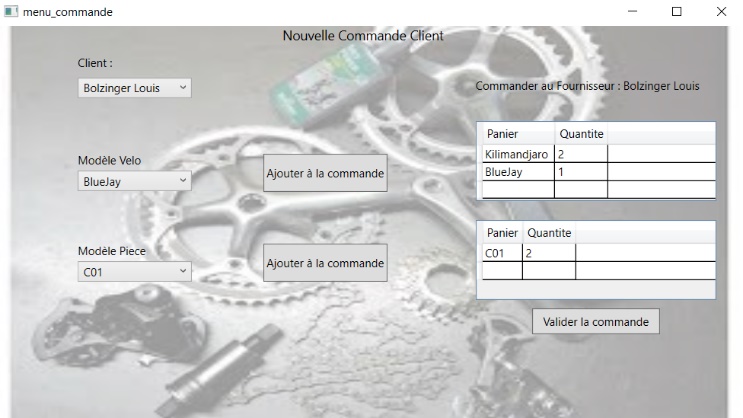
Les commandes client si elles sont correctement effectués renvoient les stock restant après la commande et le délai de livraison estimé si il manque des pièces en stock. A ce moment la, une commande fournisseur est alors nécessaire. Les commandes fournisseurs augmentent directement le stock des items en temps réel. Le prix de chaque commande est aussi indiqué à la fin de la commande

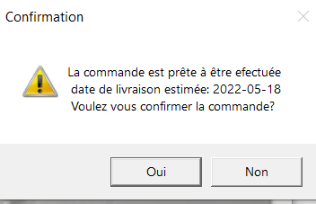
J’ai aussi proposé une innovation sur le **Contrôle de l’état d’une commande :**

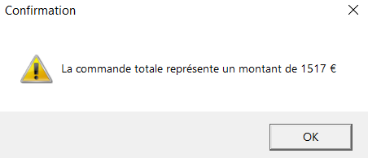
Une commande peut avoir plusiseurs statue

* En attente (le magasin à bien reçu la commande et la prépare)
* En livraison (le magasin à envoyé la commande au destinataire)
* Livré (le client à bien reçu la livraison)

 Dès la création d’une nouvelle commande un TRIGGER va déclarer le statut de la commande comme « En Livraison » étant donné qu’elle vient d’être finaliser.

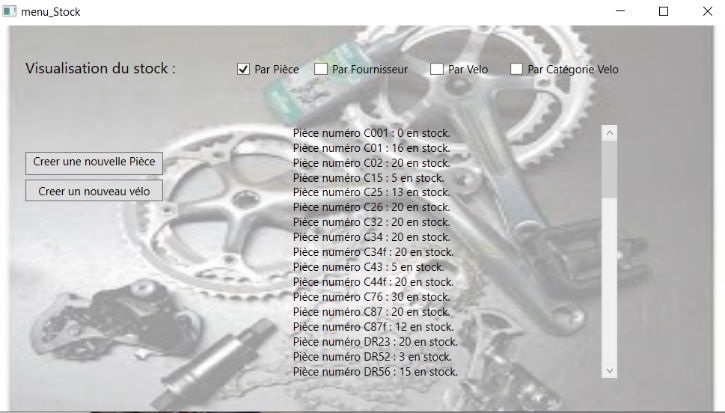








En ce qui concerne le **menu des stocks**, il est possible de visualiser l’ensemble des stock par pièce, par fournisseur, par catégorie et par velo. Attention pour un velo donné on considère que le stock Total et le stock associé au velo (vélo déjà construit) mais aussi on aditionne le stock composé des pièces correspondant à la liste d’assemblage (les vélos qui sont constructible avec le stock). La création de nouvelles pièces et de nouveaux vélos est aussi possible. Chaque nouveau vélo créer sera alors composé d’une liste de pièce qui sera décrite dans sa liste d’assemlage. Attention lors de l’arrivée dans le menu des stocks un message apparait et indique les pièces manquantes ou ne faible quantitée. Mr Lgrand peut alors les commandés au près d’un fournisseur recommandé.



Pour les menus Client et Fournisseur il est possible de créer de nouveaux, de les modifier et de les supprimer directemment en selectionnant une ligne du dataGrid. Les données des clients dont le programme Fidélio arrive à échéance dans moins de deux mois peuvent être directemment exporté en XML.

