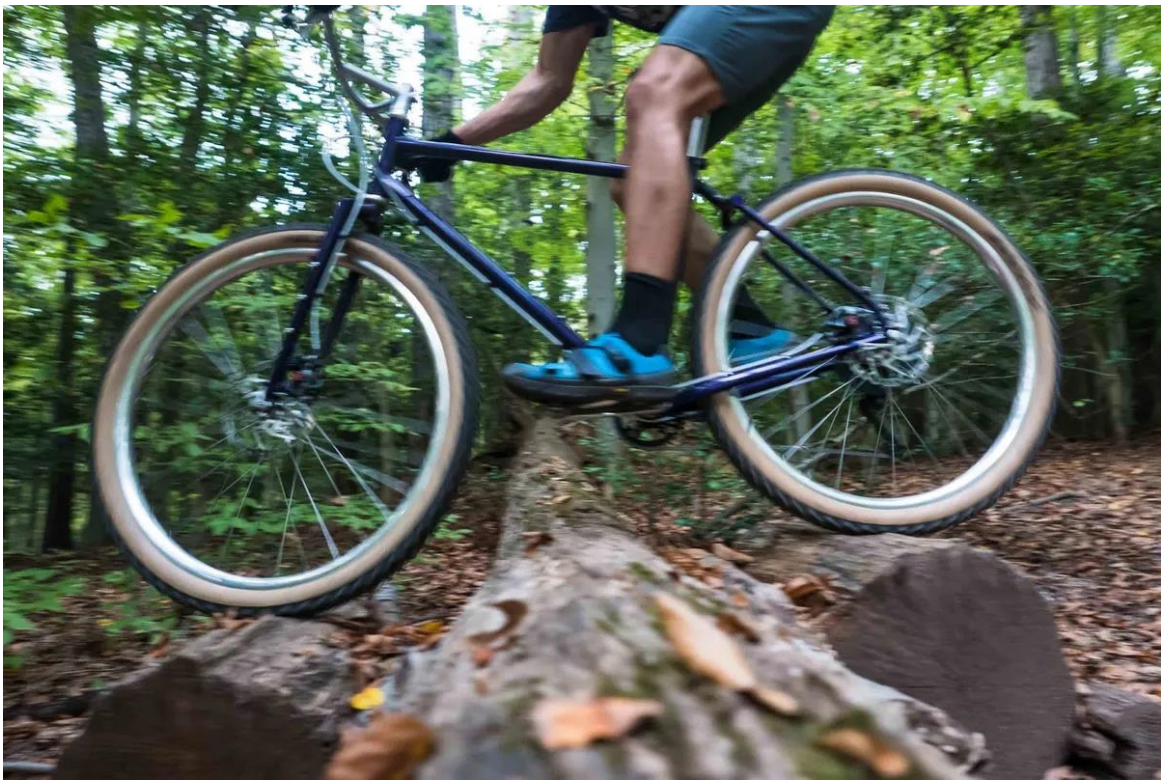


VéloMax

PROJET C# MYSQL

Janany JEGATHEESWARAN
Tania-Marie GOMES DA SILVA JUNIOR

Année 2021-2022 | Base de données & Interopérabilité | ESILV A3 | TD N

Table des matières

I. Introduction

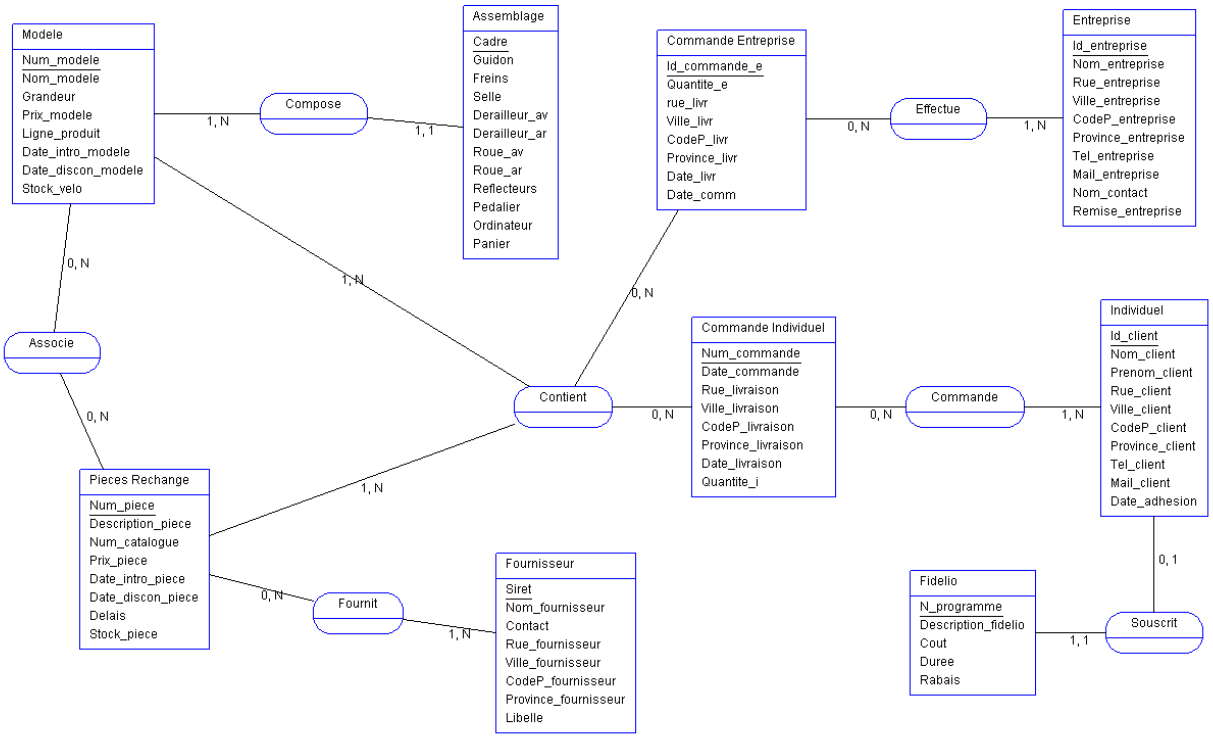
1) Problème

Max Legrand, propriétaire de VéloMax, a fait appel à nos services pour qu'on lui constitue une nouvelle base de données. En effet, il souhaite gérer au mieux ses ventes et ses clients. Plusieurs contraintes nous sont imposées au niveau des ventes, des clients et des fournisseurs. Par ailleurs, nous avons également accès à des extraits de tableau.

2) Solution

Pour répondre aux attentes de Max Legrand, nous allons créer une application de gestion de clients et de ventes de produits en C# MySql sous format WPF. L'interface est constituée de boutons permettant à l'utilisateur de gérer ses ventes, ses commandes, ses fournisseurs, ses clients et ses produits. De plus, l'utilisateur peut également demander un rapport statistique sur les items et les programmes d'adhésion.

II. Modèle E/A



Pour plus de simplicité et parce que les attributs n'ont été mentionnés que pour les pièces de rechange, nous considérons que pièces détachées = pièces de rechange.

Les modèles de vélos peuvent être constitués ou non de pièces de rechange fournies par un ou des fournisseurs et sont faits à partir d'un assemblage de pièces.

Une commande peut contenir des modèles de vélos et des pièces de rechange.

Nous avions créé au départ une entité Clientèle reliée à une entité Entreprise et une entité Individuel avec un id client qui serait utile pour les commandes. Cependant, nous avons rencontré des problèmes au niveau des dépendances donc nous l'avons supprimée.

Individuel dispose d'une date d'adhésion. Cela concerne principalement les nouveaux clients. C'est pour aider Max Legrand à connaître sa clientèle la plus récente.

Nous avons distingué les commandes Entreprise des commandes Individuel. Chacune a un id_commande propre à elle.

Ainsi, pour savoir quelles commandes sont faites par les entreprises et lesquelles par les clients individuels, chaque type de client dispose d'un identifiant (id_client et id_entreprise) qui est mentionné dans les commandes.

Pour faire la gestion des clients avec et sans programme Fidélio, le N_programme présent dans l'entité Fidélio migrera vers l'entité Individuel. Une personne n'ayant pas souscrit à un programme Fidélio aura son N_programme à 0. Il en est de même pour la remise commerciale avec les entreprises.

III. Schéma relationnel

VéloMax :

Modele (N° modele, Nom_modele, Grandeur, Prix_modele, Ligne_produit, Date_intro_modele, Date_discon_modele, Stock_velo**)**

Pieces Rechange (Num_piece, Description, Num_catalogue, Prix_piece, Date_intro_piece, Date_discon_piece, Delais, Stock_piece, #Siret, #Nom_fournisseur**)**

Assemblage (Cadre, Guidon, Freins, Selle, Derailleur_av, Derailleur_ar, Roue_av, Roue_ar, Reflecteurs, Pedalier, Ordinateur, Panier, #N°modele, #Nom_modele, #Grandeur**)**

Fournisseur (Siret, Nom_fournisseur, Contact, Rue_fournisseur, Ville_fournisseur, CodeP_fournisseur, Province_fournisseur, Libelle**)**

Commande Individuel (N° commande, Date_commande, Rue_livraison, Ville_livraison, CodeP_livraison, Province_livraison, Date_livraison, Quantite_i , #Nom_client, #N°modele, #Nom_modele, #Grandeur, #Num_piece**)**

Commande Entreprise (Id_commande_e, Quantite_e, Rue_livr, Ville_livr, CodeP_livr, Province_livr, Date_livr, Date_comm, #Id_entreprise, #Nom_entreprise, #N°modele, #Nom_modele, #Grandeur)

Individuel (Id_client, Nom_client, Prenom_client, Rue_client, Ville_client, CodeP_client, Province_client, Tel_client, Mail_client, Date_adhesion, #N_programme**)**

Fidelio (N_programme, Description_programme, Cout, Duree, Rabais**)**

Entreprise (Id_entreprise, Nom_entreprise, Rue_entreprise, Ville_entreprise, CodeP_entreprise, Province_entreprise, Tel_entreprise, Mail_entreprise, Nom_contact, Remise_entreprise**)**

IV. Implémentation sous C# MySql

1) Script SQL

On reprend les données du modèle E/A pour la création des tables au sein de la database velomax.

L'une des conditions du projet est d'avoir l'administrateur root/root et de créer un utilisateur bozo/bozo qui n'a que les droits de lecture sur la base de données :

```
#Création administrateur root-root
CREATE USER IF NOT EXISTS 'root'@'localhost' IDENTIFIED BY 'root';
ALTER USER IF EXISTS 'root'@'localhost' IDENTIFIED BY 'root';

#Création user bozo-bozo sans les privilèges
CREATE USER IF NOT EXISTS 'bozo'@'localhost' IDENTIFIED BY 'bozo';
ALTER USER IF EXISTS 'bozo'@'localhost' IDENTIFIED BY 'bozo';
GRANT SELECT, SHOW VIEW ON VeloMax.* TO 'bozo'@'localhost';
FLUSH PRIVILEGES;
```

2) Requêtes SQL spécifiques

Jointures

select f.n_programme, id_client,nom_client,prenom_client from Individuel c inner join fidelio f on f.n_programme=c.n_programme group by f.n_programme;

⇨ Cette requête permet d'afficher les clients avec leur programme Fidélio

select p.id_client,p.nom_client,count(c.id_commande_individuel), sum(quantite) from Individuel p inner join commande_individuel c on c.nom_client = p.nom_client group by p.id_client order by sum(quantite) DESC;

⇨ Cette requête permet d'afficher le « top client Individuel », c'est-à-dire le client Individuel ayant effectué le plus d'achats.

select e.id_entreprise,e.nom_entreprise,count(c.id_commande_e), sum(quantite) from Entreprise e inner join commande_e c on c.nom_e = e.nom_entreprise group by e.id_entreprise order by sum(quantite) DESC;

⇨ Cette requête permet d'afficher le « top client Entreprise », c'est-à-dire le client Entreprise ayant effectué le plus d'achats.

Valeurs moyennes

select avg(quantite) from commande_e where description_piece='0';

select avg(quantite) from commande_e where nom_modele='0';

select avg(quantite) from commande_individuel where description_piece='0';

select avg(quantite) from commande_individuel where nom_modele='0';

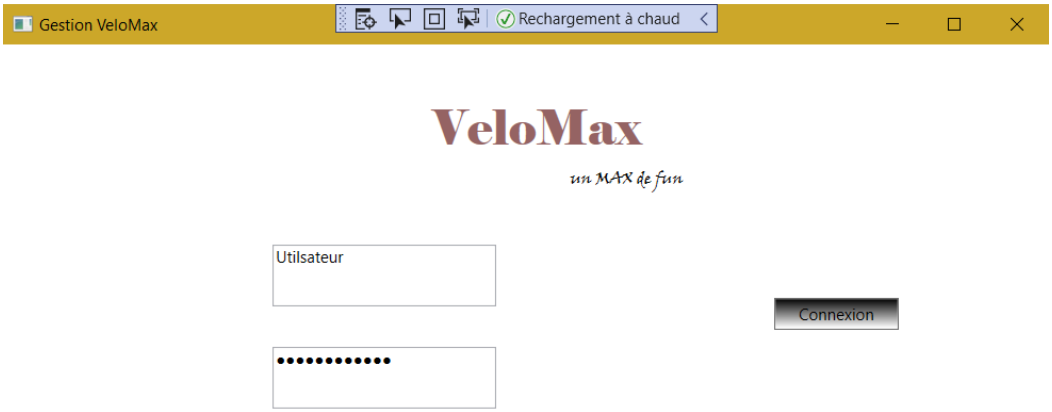
Union

select nom_client from Individuel UNION select nom_entreprise from Entreprise

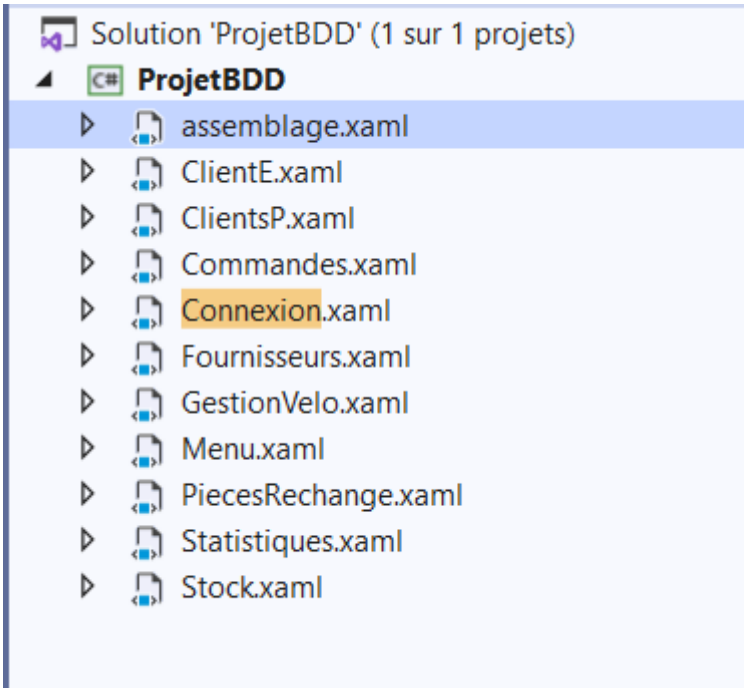
⇨ Cette requête permet d'afficher en même temps les clients Individuel et les clients Entreprise par leurs noms.

3) Implémentation WPF

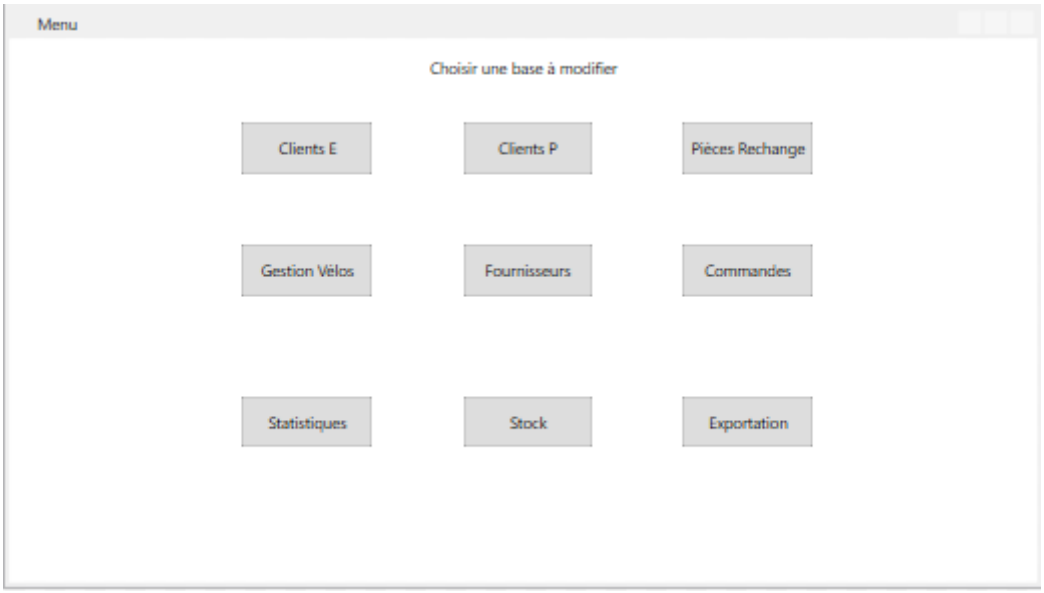
Nous avons décidé de faire notre projet sous WPF pour permettre un meilleur visuel. L'écran d'accueil se présente en tant que page de connexion pour l'administrateur root-root :



Voici les différentes pages qui permettent de naviguer sur notre application :

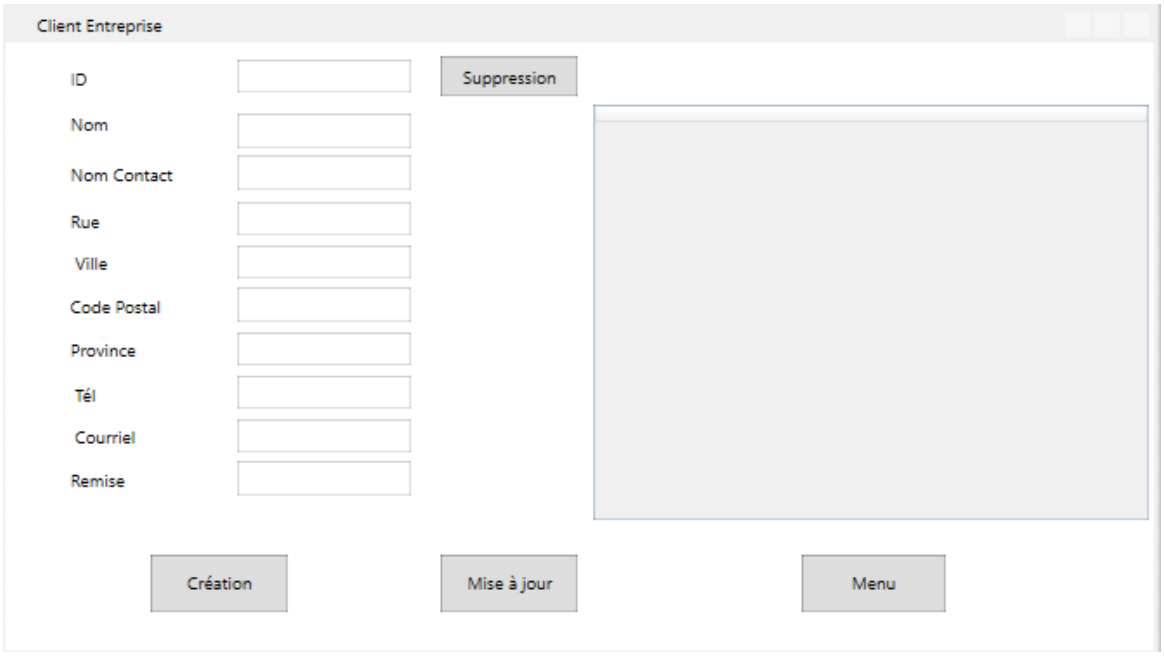


Après connexion réussie, l'administrateur tombe sur cette page où toutes les options disponibles sont présentées :

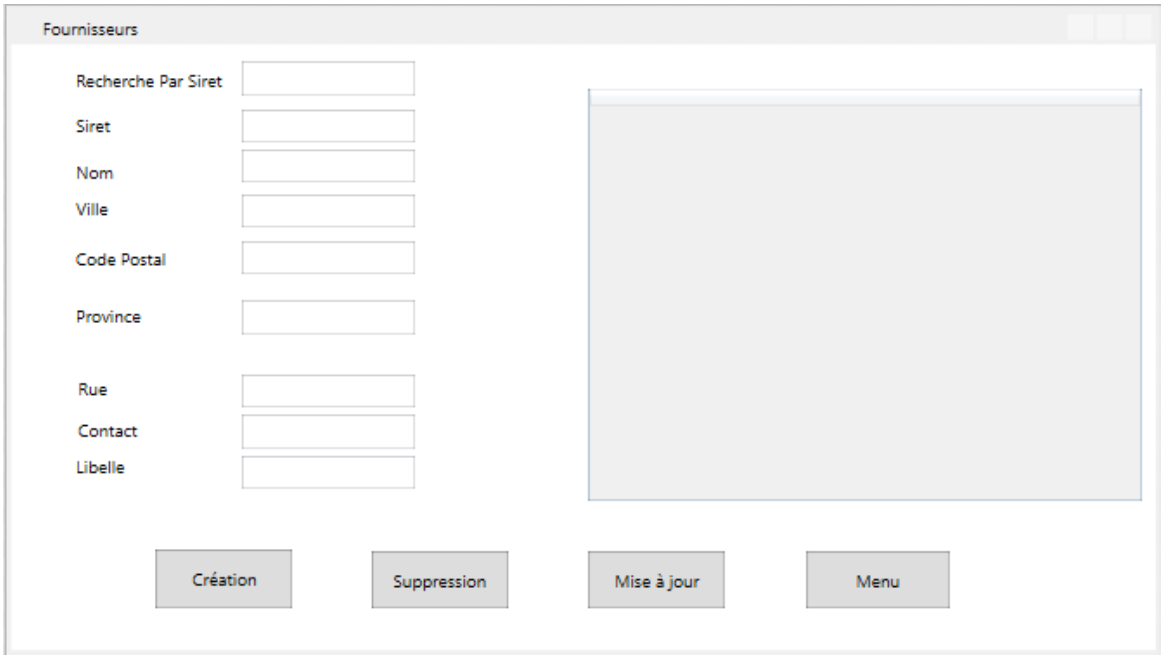


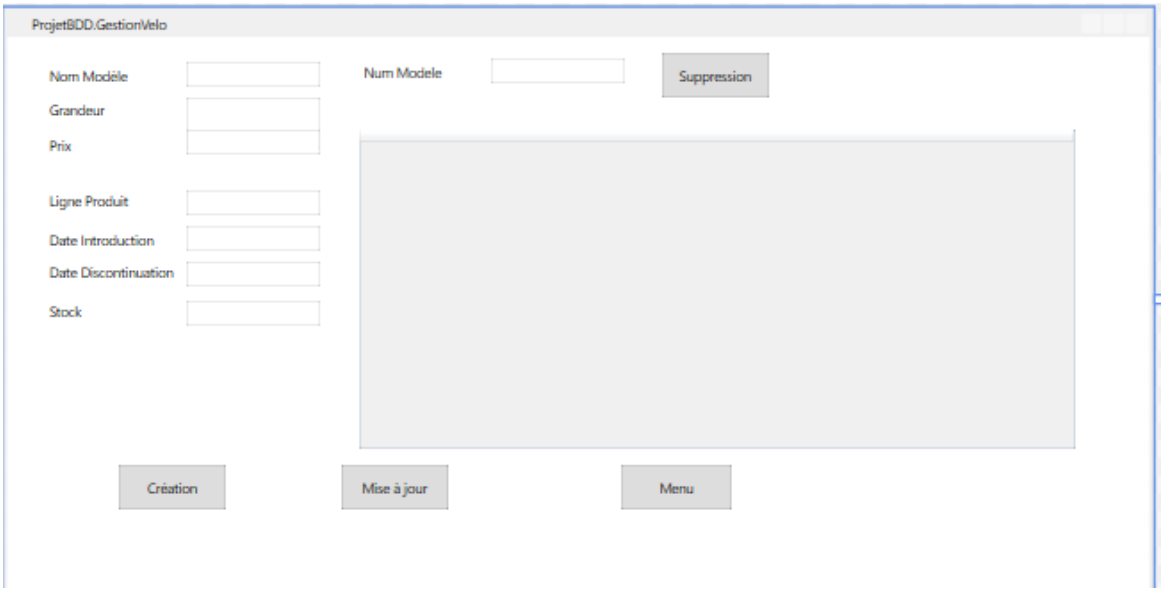
L'exportation permet d'exporter les données sous format XML ou JSON

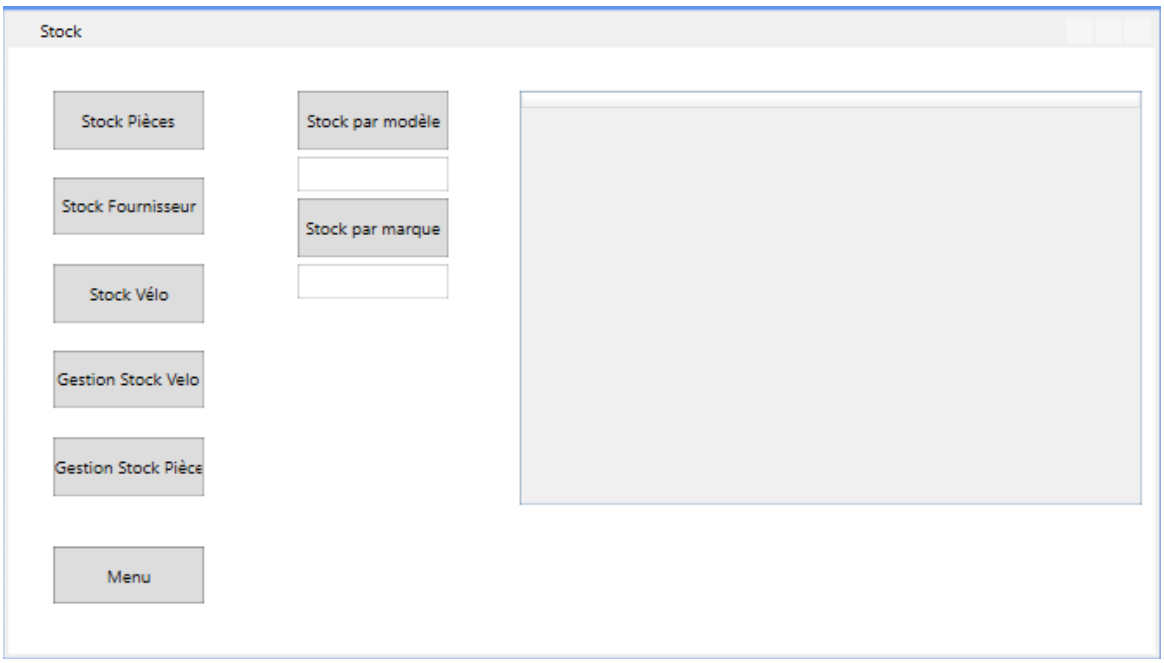
Par exemple si on souhaite consulter la base des clients Entreprise, on est sur une page où on peut renseigner les informations nécessaires pour ajouter, supprimer ou modifier. Ensuite à droit de la fenêtre, l'administrateur peut voir la base mise à jour :

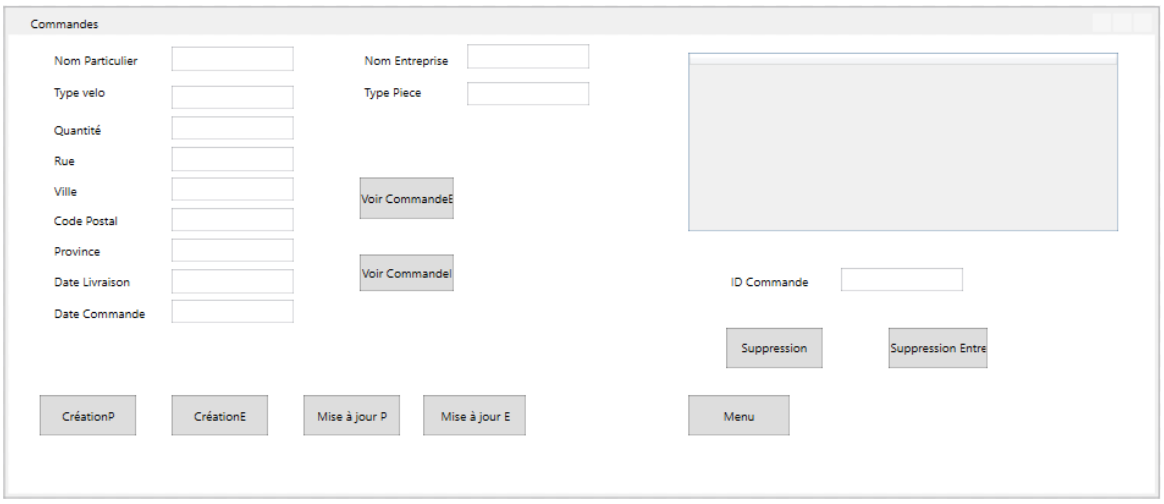


La logique et le format est le même pour les autres gestions. Ci-dessous quelques exemples des autres pages :






Stock
Stock Pièces
Stock Fournisseur
Stock Vélo
Gestion Stock Velo
Gestion Stock Pièce
Menu
Stock par modèle
Stock par marque


Commandes
Nom Particulier
Type velo
Quantité
Rue
Ville
Code Postal
Province
Date Livraison
Date Commande
Nom Entreprise
Type Piece
Voir CommandeE
Voir CommandeI
ID Commande
Suppression
Suppression Entre
CréationP
CréationE
Mise à jour P
Mise à jour E
Menu

V. En Résumé

	Réussi	Non réussi
Gestion des pièces de Rechange et vélos	X	
Gestion des clients Individuel et Entreprise	X	
Gestion fournisseurs	X	
Gestion des commandes	X	
Gestion du stock	X	
Module Statistiques	X	
Export XML/JSON	X	
Requêtes spéciales (union, jointures…)	X	
Création admin root et utilisateur bozo (sans privilèges)	X	
Connexion admin root	X	
Interface WPF	X	
Connexion utilisateur bozo		X
Alerte quand stock est faible		X