

SANCHEZ Hugo

ROCCARO Hugo

SEGUIN Lounès

## **Livrable Problème 'VéloMax'**

### Table des matières

Informations importantes .....	2
Version du logiciel.....	2
Références et Packages requis .....	2
Publication de la solution .....	2
Informations de Connexion à la Base de Données .....	2
Evolution du programme .....	2
Schéma Entité / Association .....	2
Conception de la base de données et Schémas relationnel.....	3
Réalisation du code / programmation .....	3
Ajouts et spécificités de notre programme.....	3
Affichage.....	3
Statistiques et données.....	3
Modification / Suppression des {Clients/Commandes/Fournisseurs} .....	3
Ajout de Commandes.....	3
SQL et portabilité .....	4
Conclusion .....	4

## Informations importantes

Veillez prendre connaissance de ces informations avant de faire tourner notre solution.

### Version du logiciel

Le projet a été réalisé sous Visual Studio Community 2022, avec l'option "Développement.NET Desktop" pour pouvoir créer des Windows Forms WPF.

### Références et Packages requis

Afin de réaliser ce projet à plusieurs, nous avons stocké nos données sur une base de données distantes MySQL.

- La dll MySql.Data.dll doit être ajoutée en tant que référence projet (Projet >> ajouter une référence >> sélectionner la dll)
- Il faut cliquer sur le nom de la solution dans l'explorateur de solutions >> Gérer les packages NuGet >> Parcourir, puis il faut installer les packages suivants pour que MySQL fonctionne : Renci.SshNet.Async et SSH.NET
- Concepteur de Diagramme de Classe pour visualiser un schéma UML de notre projet.

### Publication de la solution

La solution a été générée dans le dossier 'Build' de notre rendu pour pouvoir la tester sans encombre et avoir un meilleur aperçu du point de vue professionnel.

### Informations de Connexion à la Base de Données

La base de données ayant été installées sur un serveur distant créé par nous-même, il est possible d'y accéder depuis n'importe où avec les bons identifiants. C'est pour cela que nous avons pris la liberté de ne pas laisser l'utilisateur root/root pour des questions de sécurité.

~~Voici les identifiants de connexion :~~

Modifier les valeurs du fichier « Program.cs » en y mettant vos identifiants et vos paramètres de connexion à votre base de données.

## Evolution du programme

### Schéma Entité / Association

Le premier défi auquel nous avons dû faire face a été de visualiser correctement les contraintes du problème afin de modéliser la base de données. Cette étape nous a pris une semaine et nous avons dû recommencer de zéro plusieurs fois afin de répondre correctement à toutes les contraintes que nous avons identifiées.

## Conception de la base de données et Schémas relationnel

Nous avons dû réaliser le schéma relationnel à la main car les logiciels gratuits ne proposaient pas assez d'option pour tous les types possibles d'attributs ou étaient limités en nombre d'entités.

De plus, le schéma E/A et le schéma relationnel ont été modifiés au fil de l'avancement du programme face aux problèmes / contraintes supplémentaires rencontrés.

## Réalisation du code / programmation

Afin d'avoir un programme le plus proche de la réalité possible, nous avons réalisé une interface WPF avec toutes les fonctionnalités disponibles graphiquement à l'aide de ListView, button,...

Pour avoir un programme le plus clair possible, plusieurs classes ont été réalisées comme les classes Commande.cs, Client.cs, ClientParticulier.cs (hérite de Client), ClientEntreprise.cs, Fourni.cs

Chaque classe nous permet de définir nos objets principaux pour le problème et comporte une méthode AddToBDD() afin d'ajouter les éléments à la base de données.

De plus les différents Form et fichiers ont été organisés dans différents dossiers mais dans le namespace de la solution afin de mieux s'y retrouver parmi nos nombreux fichiers.

## Ajouts et spécificités de notre programme

### Affichage

Comme indiqué ci-dessus, nous avons utilisé le module Windows Form WPF pour réaliser notre interface qui se veut la plus intuitive possible.

### Statistiques et données

Certaines contraintes du livrable ont été incorporées dans le module statistiques, mais les autres points ont directement été intégrés dans les modules correspondants comme le module Client ou Commande.

### Modification / Suppression des {Clients/Commandes/Fournisseurs}

Il suffit de faire un clic droit sur la ligne de l'objet étudié pour effectuer une des différentes actions de traitement comme la suppression ou la modification de celui-ci. Nous avons trouvé que c'était la solution la plus simple et la plus intuitive.

### Ajout de Commandes

Lors de la création d'une commande, que ce soit une pièce ou un vélo, notre programme va sélectionner le meilleur fournisseur pour l'objet sélectionné et en déduire la quantité commandée. Le délai de livraison de notre entreprise est de 7 jours. Si le fournisseur ne possède pas assez de

pièces pour livrer le client, on va ajouter au délai des 7 jours de livraison le délai d'approvisionnement via le fournisseur pour l'objet commandé.

## SQL et portabilité

Nous voulons stocker nos données et pouvoir y accéder depuis n'importe où. C'est pour cela que nous avons créé une base de données distante sur un serveur linux hébergé par l'un de nous. De plus afin de pouvoir également coder à plusieurs sur le projet, nous avons ouvert une page Github privée.

La base de données (BDD) a été créée sur un serveur linux avec MariaDb tournant sur un Raspberry Pi. Un accès distant a été créé ainsi que l'ouverture des ports nécessaire pour autoriser l'accès à la BDD depuis n'importe quel endroit.

Des triggers (hors programme) ont également été réalisés dans notre base de données. Ceux-ci nous permettent lors de l'ajout d'un client qui adhère à un programme Fidelio de récupérer le délai correspondant à un programme de fidélité et d'ainsi insérer dans la table client la fin d'adhésion de celui-ci à son programme. Le 2<sup>ème</sup> Trigger nous permet quant à lui de modifier la valeur 'Nom\_Fournisseur' dans la table 'FOURNI' par le nom de l'entreprise correspondante au SIRET fournisseur inséré en tant que clé étrangère.

## Conclusion

Ce problème a été pour nous un réel défi et nous a permis de retravailler toutes les notions vues en base de données mais aussi en c#. Nous avons également réalisé qu'avec un peu plus de temps, nous aurions été capables de réaliser une véritable application de gestion pour une société de Vélo (sans prendre en compte l'aspect sécurité de notre programme), ce qui nous a nous-même impressionné.