



# RAPPORT

## PROJET VELOMAX

2021

**BDD**  
Intéropérabilité



**ALLAM** Diana **PROFESSEUR TD**

**ELLUL** Aline **RESPONSABLE MODULE**

**IZRI** Nora

**AUTEURS**

**BAUMANN** Lewis

**AIT LACEN** Souhail

**AHMED** Achirafi

**TD N**

# Table des matières

I	Présentation du Projet.....	2
II	Diagramme E/A .....	2
III	Les classes C#.cs .....	2
IV	Les classes XAML .....	3
V	Les classes XAML.cs .....	3
VI	Les Innovations.....	4
VII	Interface Homme-machine.....	4
VIII	Conclusion et Critiques.....	4

## I Présentation du Projet

Dans le cadre du problème de BDD, nous devons réaliser en monôme ou en binôme ou en trinôme, une interface en WPF permettant de simuler la création de la gestion d'un magasin de vélo et donc créer les méthodes et classes permettant cela. Il fallait aussi créer une base de données en SQL permettant de répondre aux demandes du cahier des charges.

Bien sûr, il fallait aussi créer des innovations et au préalable créer et construire un diagramme E/A.

Ainsi, l'objectif de ce rapport est de présenter les différentes méthodes et classes (XAML et .cs) que nous avons créé en lien avec le projet mais aussi faire part des problèmes que nous avons rencontrés.

Dans chaque section et/ou sous-section, nous pourrions utiliser des noms d'attributs associés aux classes appelées, ainsi veuillez les retrouver dans le programme C# ou bien dans la base de données SQL.

## II Diagramme E/A

Avant de commencer, expliquons ce que signifie un diagramme E/A. C'est un modèle de données ou diagramme pour des descriptions de haut niveau de modèles conceptuels de données. Il fournit une description graphique pour représenter des modèles de données sous la forme de diagrammes contenant des entités et des associations. De tels modèles sont utilisés dans les phases amont de conception des systèmes informatiques.

Ils sont utilisés, par exemple, pour décrire les besoins en information et/ou le type d'information qui doit être enregistré dans les bases de données pendant la phase d'élaboration du cahier des charges.

L'E/A doit être donc réalisé avant le commencement de la programmation sur Visual Studio et Workbench, ce diagramme nous permet de savoir comment sera constituée notre base de données. **La visualisation de ce diagramme est disponible en annexe.** Dans ce diagramme vous pouvez apercevoir les différentes tables (Clients, assemblage, car-fidelio, commande, feat (cette table est spéciale car elle crée sur Visual Studio elle contient les éléments du JSON), fidelio, fournisseur, itemcommande, piecefournisseur, pièces et vélo).

Nous allons donc passer à la partie code et son explication dans les parties suivantes.

## III Les classes C#.cs

Dans cette partie nous allons parler de deux classes créées qui sont utilisées pour seulement deux choses, l'export en XML et JSON.

Pour sérialiser, nous devons avoir des classes propres pour cela, Velo et Clients sont créées pour cela. Nous avons, pour certains attributs, utilisé [JSONPROPERTY] pour que seulement certaines informations apparaissent sur le fichier JSON. De même pour le fichier XML.

## IV Les classes XAML

Nous avons un total d'environ **14** pages **XAML**.

Dans ces classes, nous décrivons le visuel de ce que l'utilisateur verra, en lien direct avec les classes **XAML.cs**. Il fallait donc apprendre à manipuler les **Textblock**, **Textbox**, **MessageBox** ou les **Buttons**. Créer notre charte graphique qui sera notre base pour toutes les pages que l'on va créer a été très difficile, nous avons rencontré de nombreuses erreurs. Notamment à cause de la mise en page et de la structure de nos **Grid** pour bien placer nos **buttons** ou **images** par exemple.

Nous avons aussi utilisé d'autres bibliothèques, nous permettant par exemple selon un temps donné d'afficher les pages, grâce à **Frame** mais aussi à **System.Windows.Media.Animation** par exemple, ce qui nous permet de créer des animations afin d'apporter notre touche de **créativité**.

Une fois notre structure acquise, nous l'avons réutilisée majoritairement pour toutes les autres pages (hormis par exemple la page statistique).

## V Les classes XAML.cs

Dans cette partie, outre quelques attributs rajoutés pour permettre de les utiliser dans quelques méthodes, nous implémentons les méthodes en lien avec les **Buttons**, mais aussi des méthodes en lien avec l'utilisation des requêtes **SQL** nous permettant de répondre à des besoins donnés.

## VI Requêtes SQL

Parmi les requêtes certains types nous ont été imposés, comme une requête de sélection, une auto-jointure et une union. Concernant cette dernière, elle est présente à 2 reprises notamment dans la partie statistique. Exemple pour avoir la moyenne du prix des commandes on a : **"SELECT avg(prixTotal) from clients as c, commande , (SELECT sum(prix) as prixTotal, No\_commande as numCommande FROM( SELECT p.prix, i.No\_commande FROM pieces as p, itemcommande as i where p.numpiece = i.Ref\_item UNION ALL SELECT v.prix, i.No\_commande FROM velo as v, itemcommande as i where v.numProduit = i.Ref\_item and i.Ref\_item) as sub GROUP BY numCommande) as p WHERE numCommande = commande.No\_commande and commande.No\_client = c.No\_client"**.

On peut d'ailleurs remarquer que cette requête fait également appel à de la sélection. Quant à l'auto-jointure elle est énormément présente dans la partie commande par exemple avec des auto-jointure entre commande et itemcommande et autre. Exemple : **"Select c.No\_commande, cl.nom, prenom, Ref\_item, v.nom, v.prix, date\_commande, date\_livraison from commande c, clients cl, itemcommande i, velo v where c.No\_client=cl.No\_client and i.No\_commande=c.No\_commande and v.numProduit=i.Ref\_item order by c.No\_commande"**.

De plus, étant un groupe de 3, des requêtes non vues en classe nous ont été demandées. Pour satisfaire cette condition, nous avons fait 1 procédure et 2 triggers dont une qui fait appel à la procédure. D'ailleurs, celui-ci a pour objectif de réguler le stock en fonction des commandes. Pour ce

faire, il fait appel à la procédure qui lui permet de décrémenter les pièces dans le stock. Quant au premier trigger, il concerne la table commande où si on ne rentre pas de date de livraison alors on associe à date de livraison une date qui correspond à 1 an de plus que la date de passage de commande. Nous pouvons par la suite après analyse du stock modifier la date de livraison.

## VII Les Innovations

Comme demandé par l'énoncé du projet et son cahier des charges nous devons réfléchir à des innovations distinctes, les innovations sont les suivantes :

- 1) L'ajout d'une musique tout au long du projet, avec un ajout d'un bouton permettant de couper ou de relancer la musique à tout moment (dès le départ elle est en muet).
- 2) L'ajout d'un menu en forme d'onglets mais aussi la modification des boutons réduire et fermer qui permettent de réduire ou de fermer la fenêtre de l'application à tout moment.
- 3) L'ajout d'un modèle d'animation permettant d'afficher les pages progressivement de manière fluide, ainsi que l'ajout d'un bouton « jouer la démo » qui affiche progressivement, avec des boutons précédant et suivant, ce qui est demandé.

## VIII Interface Homme-machine

Dès le début de l'ouverture du projet l'utilisateur a le choix sur l'envie ou pas d'écouter de la musique puis nous pouvons afficher ce qui est contenu dans le MainWindow.XAML, ce qui suit l'activation selon les désirs et actions de l'utilisateur, de la liaison ou non des autres pages. Tout le projet est donc en WPF comme suggéré par le cahier des charges.

Un des objectifs principaux lors de la réalisation du WPF a été de réaliser une interface claire et intuitive, afin d'alléger au maximum la charge de l'utilisateur.

## IX Conclusion et Critiques

Ce projet nous a permis de prendre conscience de certaines de nos faiblesses, mais a également permis de nous dépasser. Une gestion intelligente de notre temps, des réunions régulières ainsi que l'émergence quasi-naturelle d'un esprit de compétition sain ont donné naissance à un projet dont nous sommes fiers.

Cependant, des points d'amélioration subsistent : des brainstorming plus réguliers, un design plus performant de l'application et des informations contenues dans la base de données nous auraient permis d'être encore plus efficaces.

Nous avons énormément appris durant la réalisation de ce projet, tant sur l'aspect technique qu'organisationnel. C'est donc avec beaucoup d'émotions que nous clôturons ce projet.