

Rapport BDD A3 - 2022

Iman TOURRES, Victor TARROUX, Tom XIE

[repo](#) [github](#)

Prévisualisation

VeloMax

Search...

Exit

Dashboard

Bikes

Parts

Clients

Suppliers

Orders

Ordered

Settings

Dashboard

Trail Explorer

DV132

Sold : 7

Quantity : 3

Price : 525

Average stats

Table

Order

Stock

Other

VeloMax

Search...

Exit

Dashboard

Bikes

Parts

Clients

Suppliers

Orders

Ordered

Settings

Bike

#	Name	Target	Price (\$)	Type	
101	Kilimandjaro	Adultes	569	VTT	
102	NorthPole	Adultes	329	VTT	
103	MontBlanc	Jeunes	399	VTT	
104	Hooligan	Jeunes	199	VTT	
105	Orléans	Hommes	229	Vélo de course	
106	Orléans	Dames	229	Vélo de course	
107	BlueJay	Hommes	349	Vélo de course	
108	BlueJay	Dames	349	Vélo de course	
109	Trail Explorer	Filles	129	Classique	
110	Trail Explorer	Garçons	129	Classique	
111	Night Hawk	Jeunes	189	Classique	
112	Tierra Verde	Hommes	199	Classique	
113	Tierra Verde	Dames	199	Classique	
114	Mud Zinger I	Jeunes	279	BMX	
115	Mud Zinger II	Adultes	359	BMX	

Add

Info / Edit

Delete

Technologie

Avalonia

Les trois membres étant sur Linux, WPF n'était pas envisageable car pas disponible sur Linux, on a donc utilisé une dérivé qui est **Avalonia**

Problèmes rencontrés

Langage

On a pas eu beaucoup de cours de C# car nous 3 on a pris le module C/C++ pendant le 1er semestre, donc on a vu le C# que pendant le module de Complément de formation.

Documentation

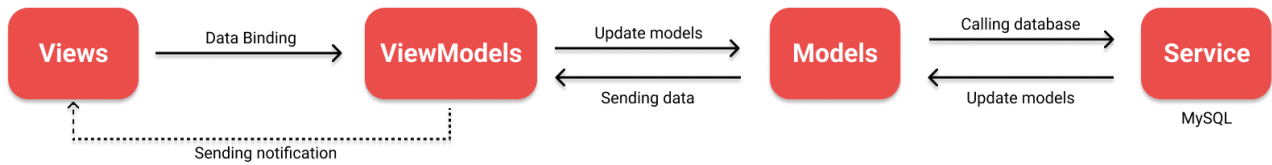
La documentation sur Avalonia étant limité, c'était difficile de tout comprendre afin de concrétiser nos idées. Du coup on a pris beaucoup de temps à comprendre et lire la documentation.

Temps

Avec la contrainte du temps et de la superposition de plusieurs autres projets, on a eu peu de temps pour réaliser ce projet.

Partie technique

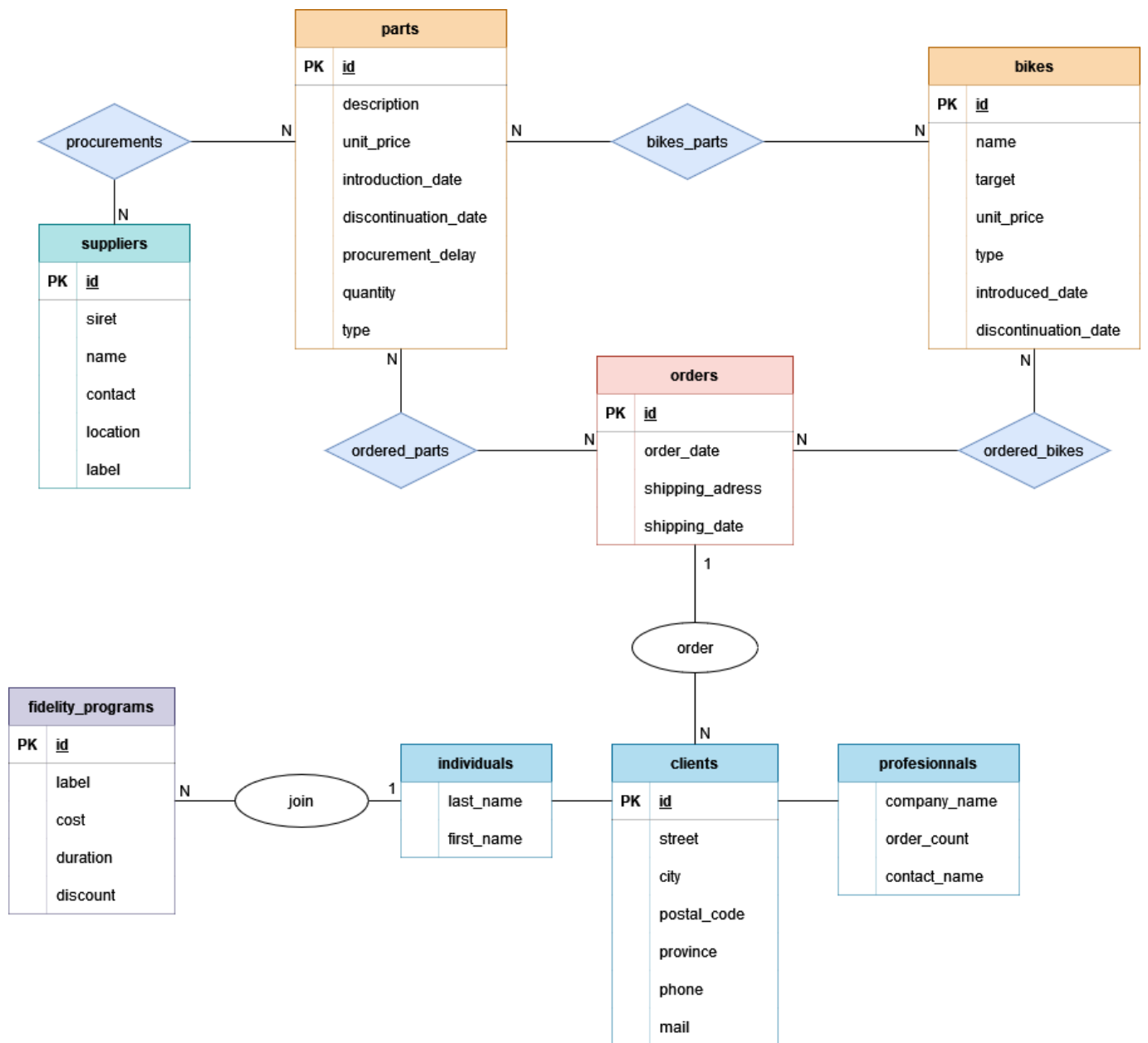
MVVM Patron de conception



On a utilisé une dérivée du patron de conception ****MVC**** (model-view-controller) pour implémenter notre application, la plupart des projets WPF implémentent le MVVM (model-view-viewmodel), on a donc décidé de suivre ainsi cette structure pour organiser les fichiers.

SQL

Modèle MCD



Exemples de requêtes utilisées

1. *Connaitre la piece la plus vendu, le benefice induit par cette piece et la quantité moyenne vendu par commandes de cette piece*

```
SELECT P.description , sum(OP.quantity), sum(OP.quantity)*P.unit_price,  
avg(OP.Quantity) FROM ordered_parts OP JOIN parts P ON OP.parts_id=P.id  
JOIN orders O ON OP.orders_id = O.id  
WHERE O.shipping_date>' " + year + "-" + mois + "-01'  
GROUP BY P.id  
HAVING sum(OP.quantity)>= all(SELECT sum(quantity) FROM ordered_parts GROUP BY  
parts_id);
```

2. *Pouvoir acceder a toutes le pieces en fonction du `string` mis dans la fonction csharp*

```
SELECT B.string, P.description, P.Quantity  
FROM bikes B JOIN bike_parts Br ON B.id=Br.bikes_id  
JOIN parts P ON P.id=Br.parts_id  
ORDER BY B.string;
```

3. *Pouvoir avoir le prix moyen d'une commande, la quantité moyenne de pieces vendus ainsi que la quantité moyenne de velos vendus*

```
Select avg(Bp.quantity*B.unit_price + Pp.quantity * P.unit_price), avg(Bp quantity),  
avg(Pp.quantity)  
From orders O JOIN ordered_bikes Bp ON O.id=Bp.orders_id  
Join bikes B on B.id = Bp.bikes_id  
join ordered_parts Pp on Pp.orders_id = O.id  
join parts P on P.id = Pp.parts_id;
```

Les livrables

Tâche	Réalisé	Partiellement réalisé	Non réalisé
IHM (interface homme machine), Avalonia	X		
Gestion pièces de rechange (création, suppression, maj)	X		
Gestion des clients (particuliers et professionnels)	X		
Gestion des fournisseurs	X		
Gestion des commandes			X
Profil de connexion à la BDD	X		
Affichage des stocks	X		
Vérification des champs de saisies lors de la création/modification	X		
Export des données (JSON/XML)	X		
Statistiques		X	
Bonus : barre de recherche sur toutes les tables	X		