

Activité_partie 3

Temps d'attente moyen pour être corrigé sur cet exercice : 52 jours

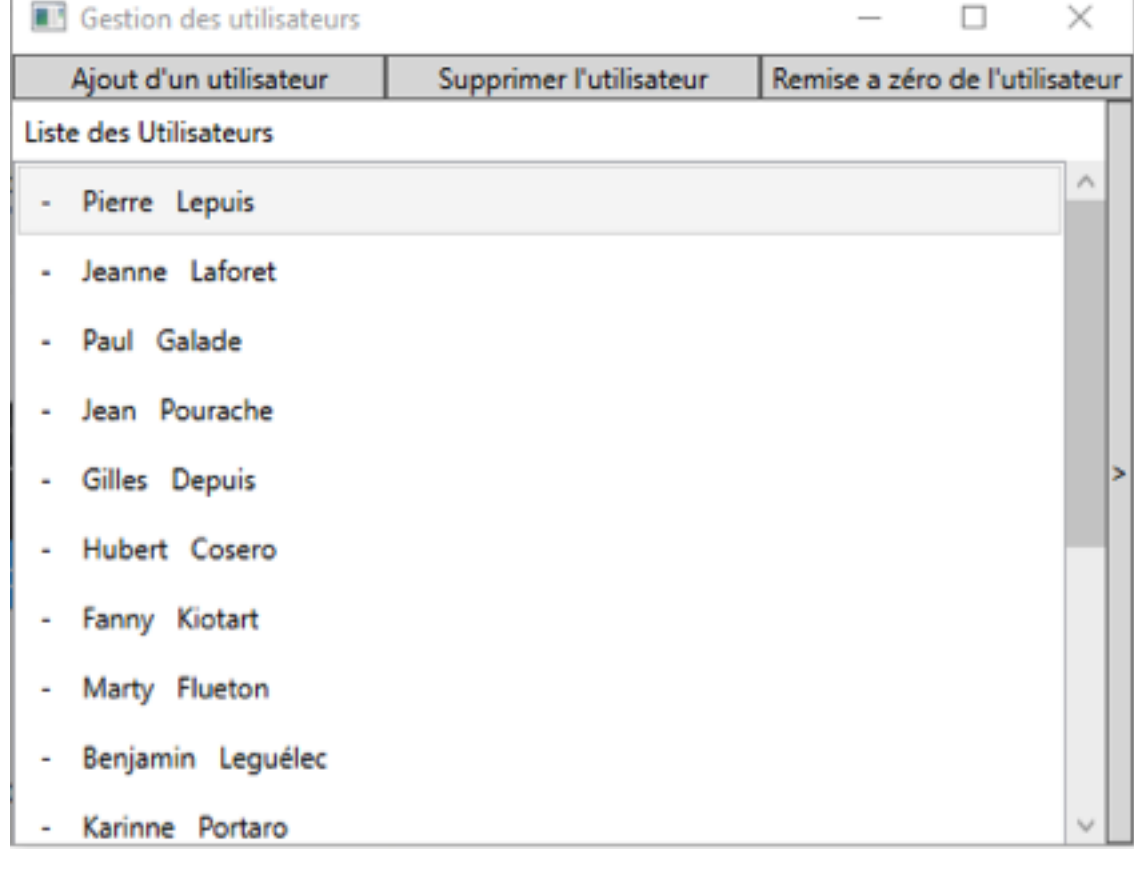
← Retour au cours

Vous savez maintenant comment réaliser une interface riche en composants graphiques fonctionnels WPF et interagissant avec un modèle (pour l'instant éphémère) de données. Vous devez avoir saisi l'intérêt d'utiliser le design pattern MVVM pour séparer clairement l'interface, des fonctionnalités offertes par cette dernière et de l'accès aux données. Pour vérifier votre aisance à manipuler ces technologies et concepts, je vous propose d'essayer à partir de zéro de réaliser l'application suivante.

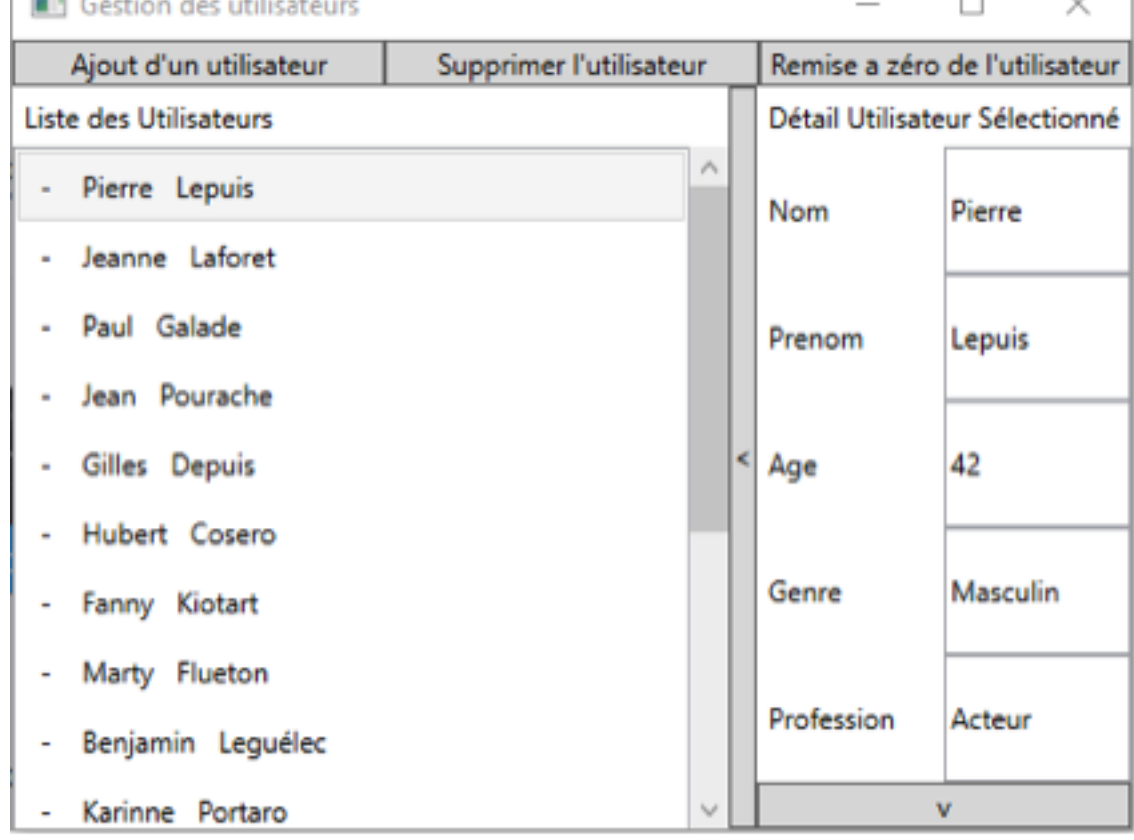
L'application devra permettre la manipulation d'un ensemble d'utilisateurs et de pouvoir modifier les données de ces derniers. La grande différence avec notre précédente réalisation est que l'agencement des éléments dans notre application sera dynamique.

En effet, votre application doit proposer 3 agencements différents :

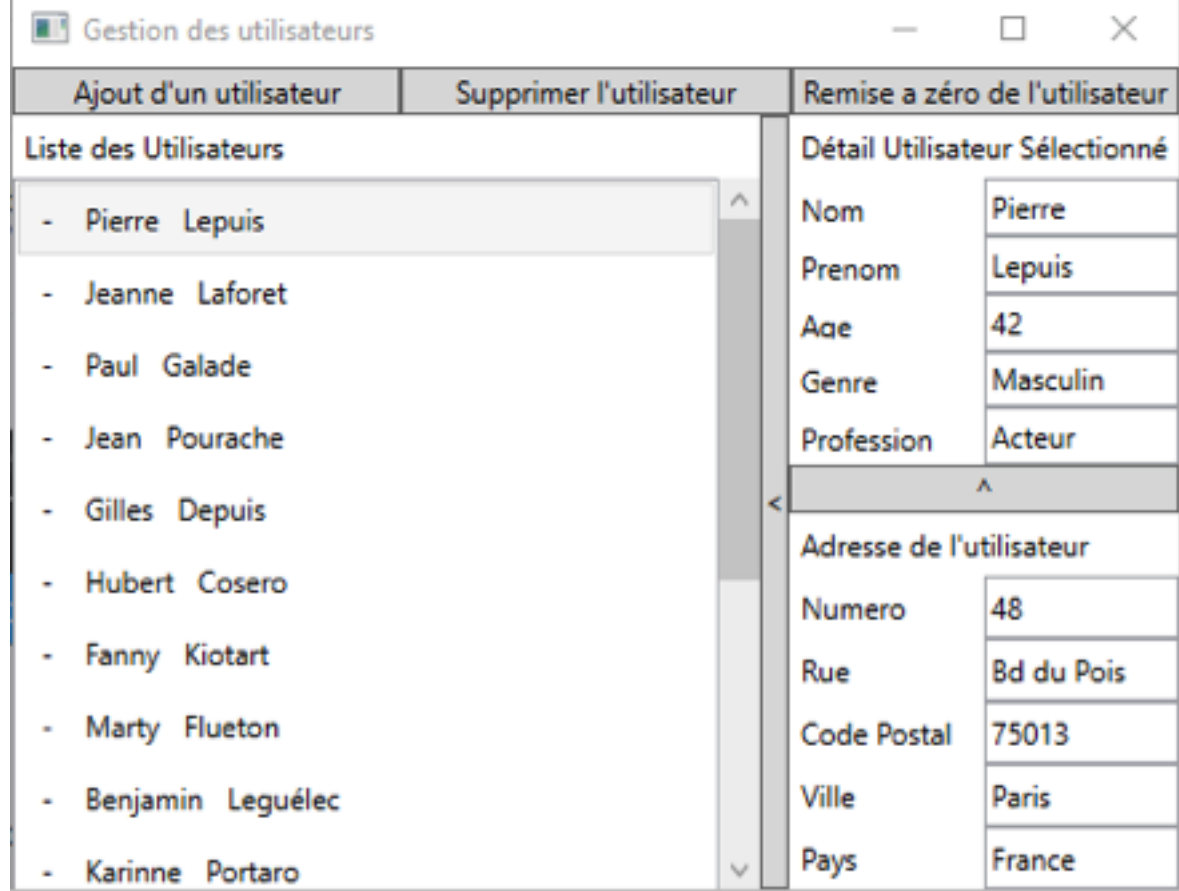
- Vue de la liste des utilisateurs :



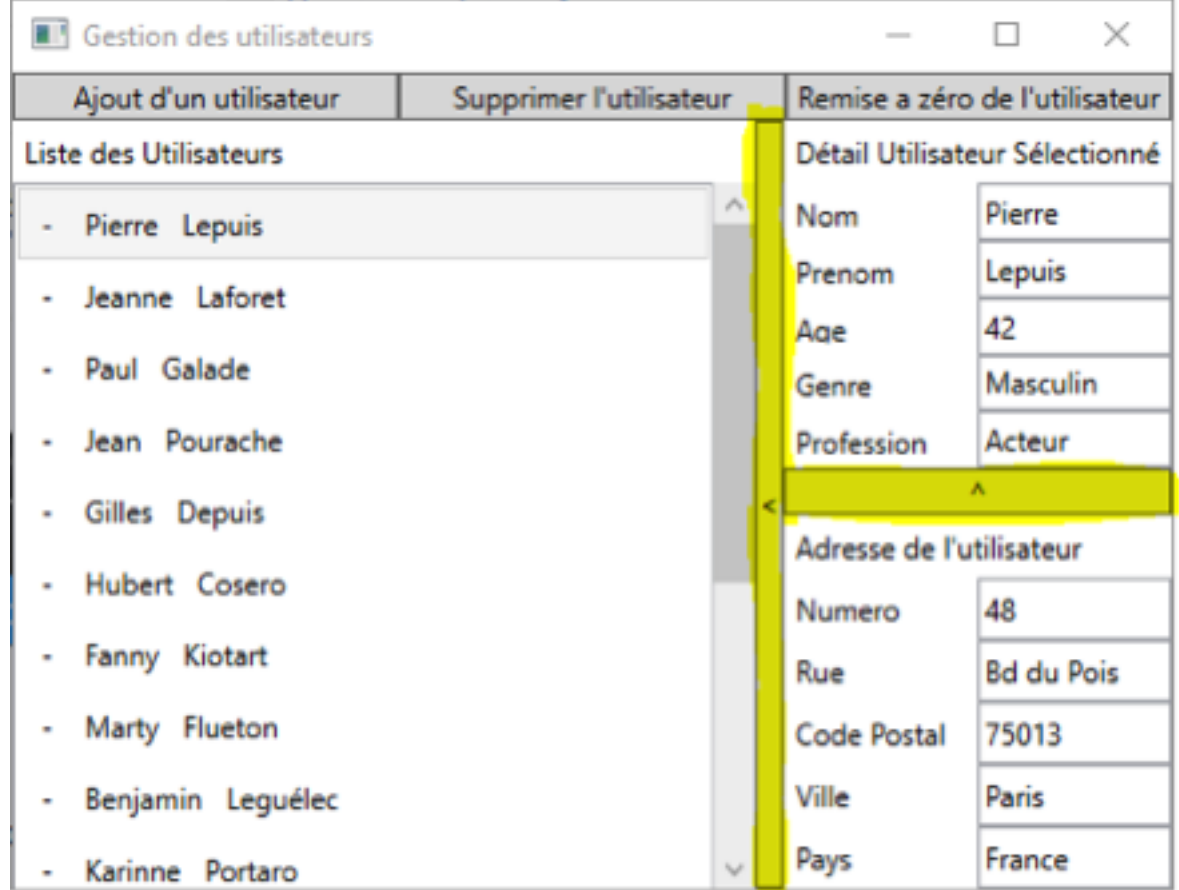
- Vue de la liste des utilisateurs et du premier niveau de détail de l'utilisateur sélectionné :



- Vue de la liste des utilisateurs, du premier niveau de détail de l'utilisateur sélectionné ainsi que de son deuxième niveau de détail (ici son adresse) :

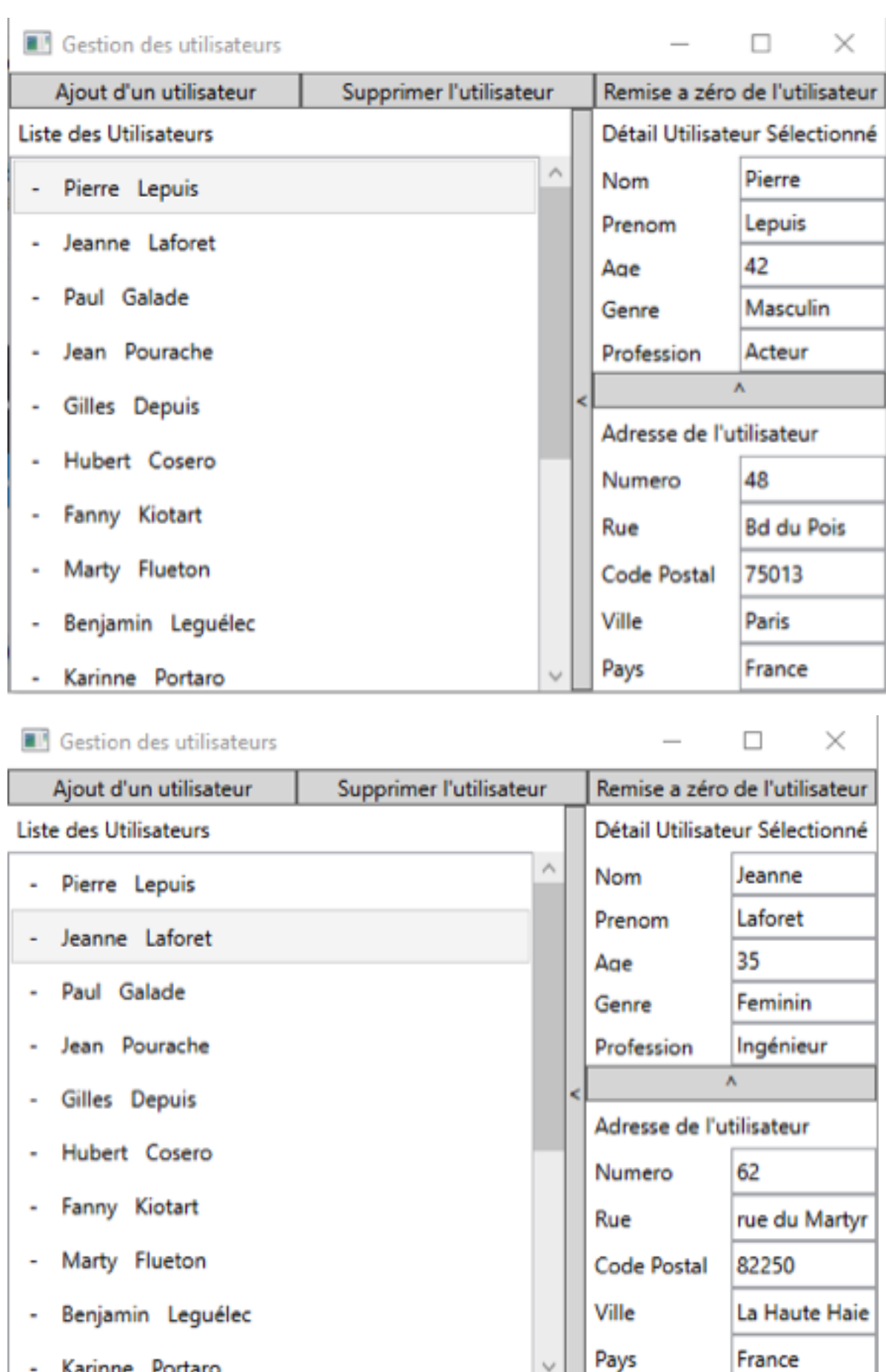


Le passage d'un agencement à un autre doit se faire à l'aide des deux barres horizontales et verticales mises en surbrillance ci-dessous :



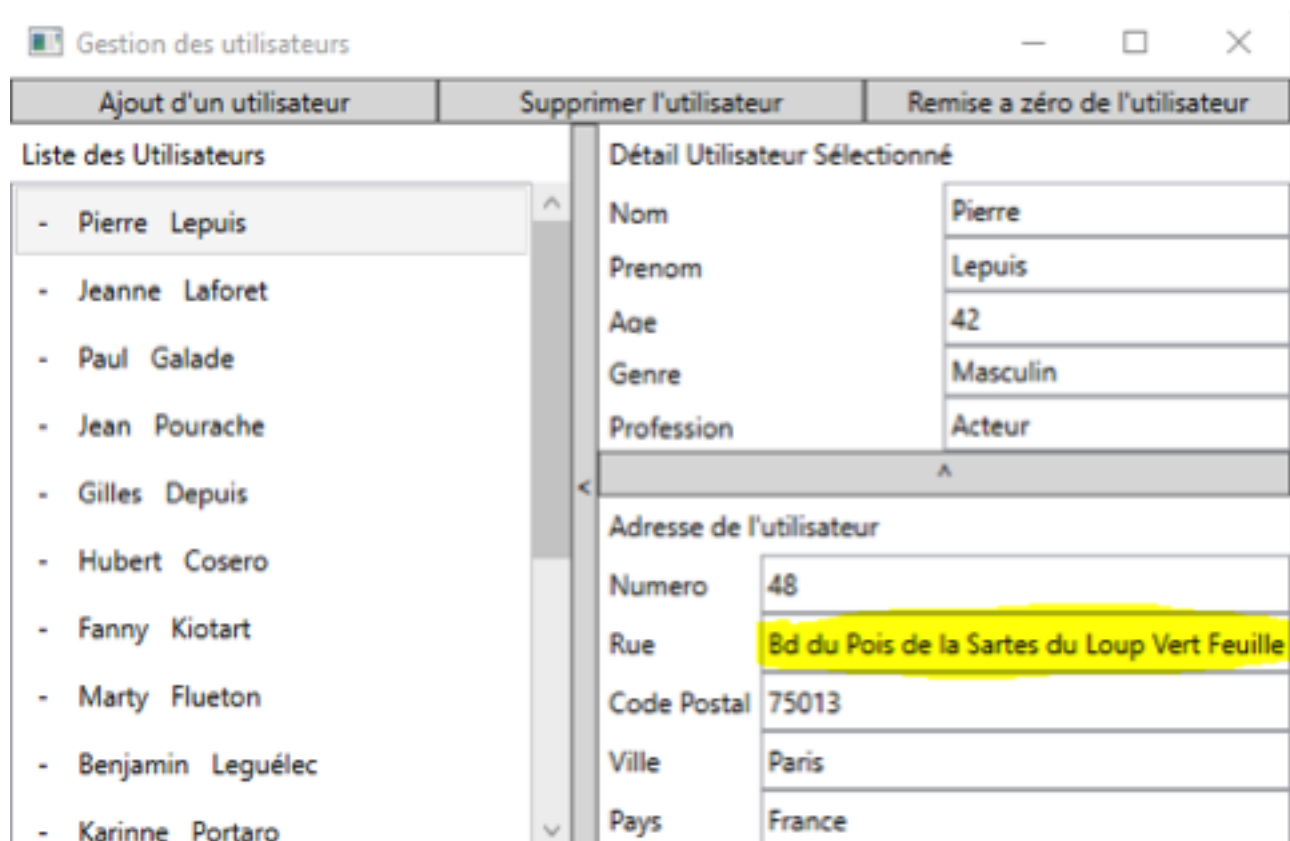
Les fonctionnalités traditionnelles d'ajout, de suppression et de remise à zéro d'un utilisateur doivent être présentes.

La sélection d'un utilisateur dans la liste doit immédiatement mettre à jour les deux panneaux de détails s'ils sont ouverts (ne plus utiliser un bouton spécifique comme dans l'exercice précédent) :

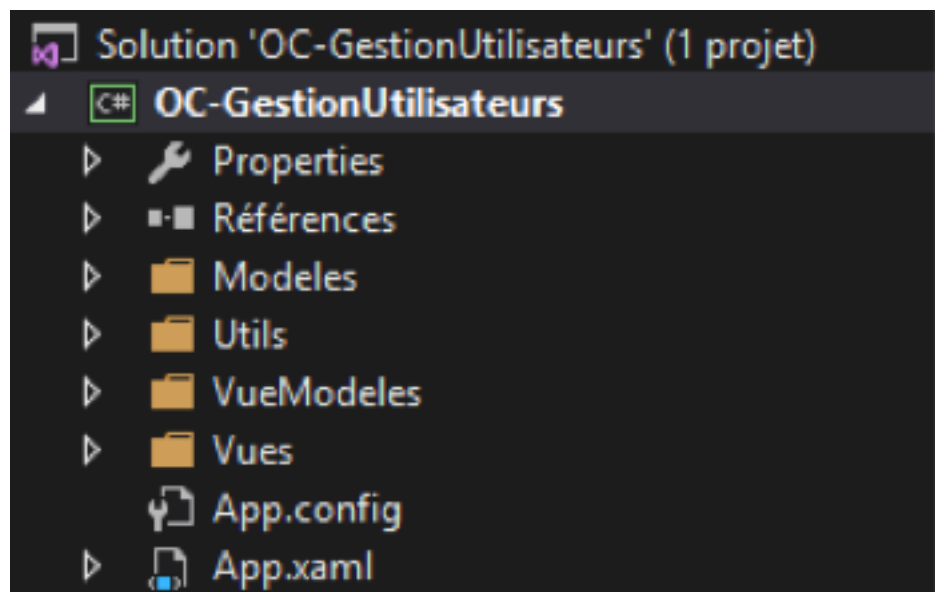


Les modifications sur les paramètres d'un utilisateur doivent être persistantes tout au long de la durée de lancement du programme.

L'application doit être robuste au changement de taille, les textes ne doivent pas être coupés dans la mesure du possible :



Concernant l'architecture de votre projet et code source, il devra être clairement identifié ce qui relève de la vue, du vue-modèle, et du modèle :

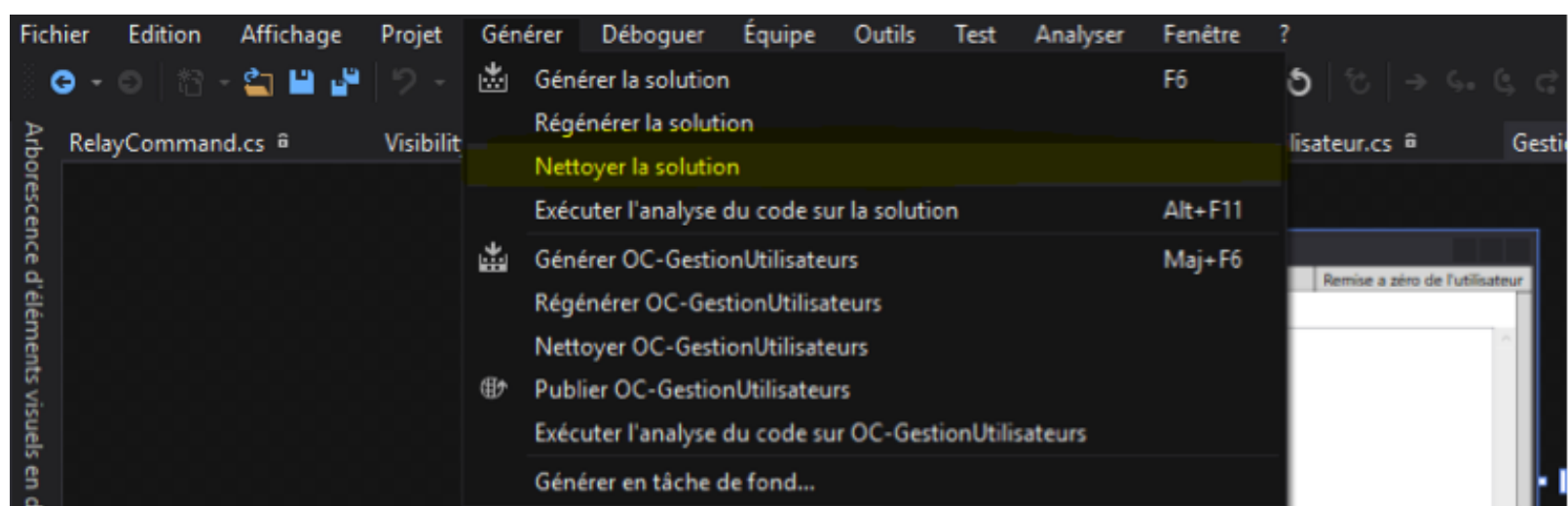


Votre code devra être propre, concis, et respecter les notions vues précédemment à savoir :

- La vue n'interagit avec le vue modèle que via Databinding sur ce dernier et invocation de Commandes.
- Le vue modèle ne fait pas référence à la vue.
- Le modèle ne fait référence ni à la vue, ni au vue modèle.

Si certaines sections de votre code sont complexes, n'hésitez pas à laisser des commentaires pour faciliter la correction.

Lorsque vous pensez en avoir terminé, et après avoir suffisamment testé votre programme, commencez par Nettoyer l'ensemble de votre solution Visual Studio (inutile de fournir les fichiers temporaires et exécutables à votre correcteur) :



Ensuite, réalisez une archive zip de toute votre solution Visual Studio

Une fois votre archive zip faite, envoyez là à l'aide de votre formulaire ci-dessous.

Vous serez évalués sur :

- La compilation de votre code :
 - Absence d'erreur
 - Absence d'avertissement
- La simplicité et la clarté de votre code :
 - Indentation
 - Méthodes bien nommées
 - Variables explicites
 - Concision du code
- Le respect des consignes :
 - Toutes les fonctionnalités et spécifications énoncées plus haut seront requises

N'hésitez pas à laisser des commentaires dans votre code pour aider le correcteur dans son travail.

Si vous avez des doutes sur comment compresser vos fichiers (c'est-à-dire, créer un dossier .zip), visionnez les vidéos indiquées dans la partie « Sélection du travail ».

Allez, c'est parti !

Sélection du travail

Votre travail (format .zip, 70 Mo max)

Choisir un fichier Aucun fichier choisi

Comment créer un fichier ZIP sous Windows ? Sous Mac OS X ?

Vous pouvez laisser un mot à l'attention de vos correcteurs si vous le souhaitez :

(facultatif)

Validation

Attention Relisez-vous bien ! Après soumission de votre travail, vous ne pourrez plus le modifier ni en renvoyer un nouveau !

Valider