TP 3 – Mon Smart Agenda Avancé

Objectif du TP :

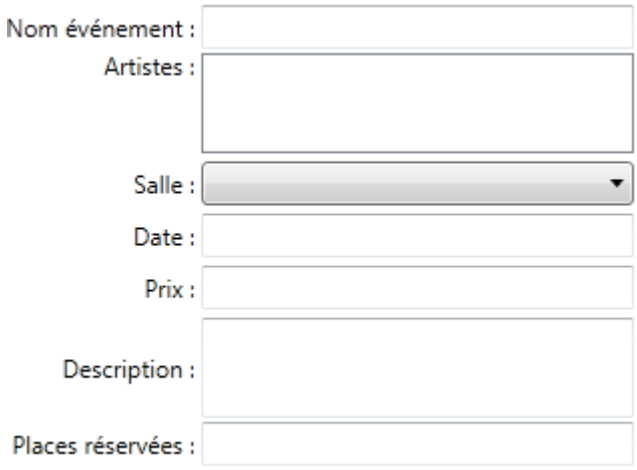
Enrichir l'application WPF du TP2. Créer un UserControl, visualiser et pouvoir ajouter des évènements, en utilisant le pattern MVVM. Dans cette optique, deux fichiers sources seront fournis et intégrés au projet :

RelayCommand.cs

ViewModelBase.cs

Etape 1 : Création d'un contrôle personnalisé

Créer un UserControl (WPF) que vous nommerez PlanningElementView. Il permettra d'afficher les propriétés d'un planning élément. Il pourra ressembler au modèle ci-dessous.

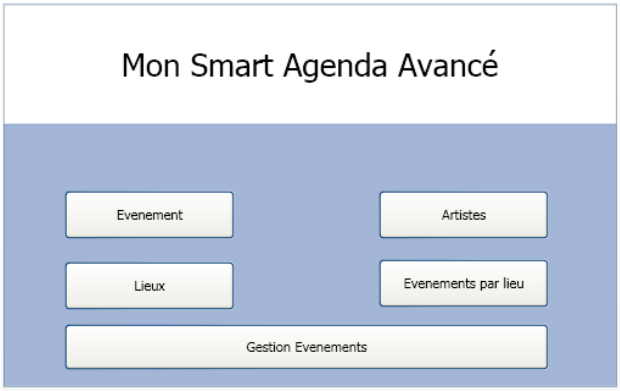


Penser à gérer le dimensionnement de ce control. Il ne devra contenir aucun code métier.

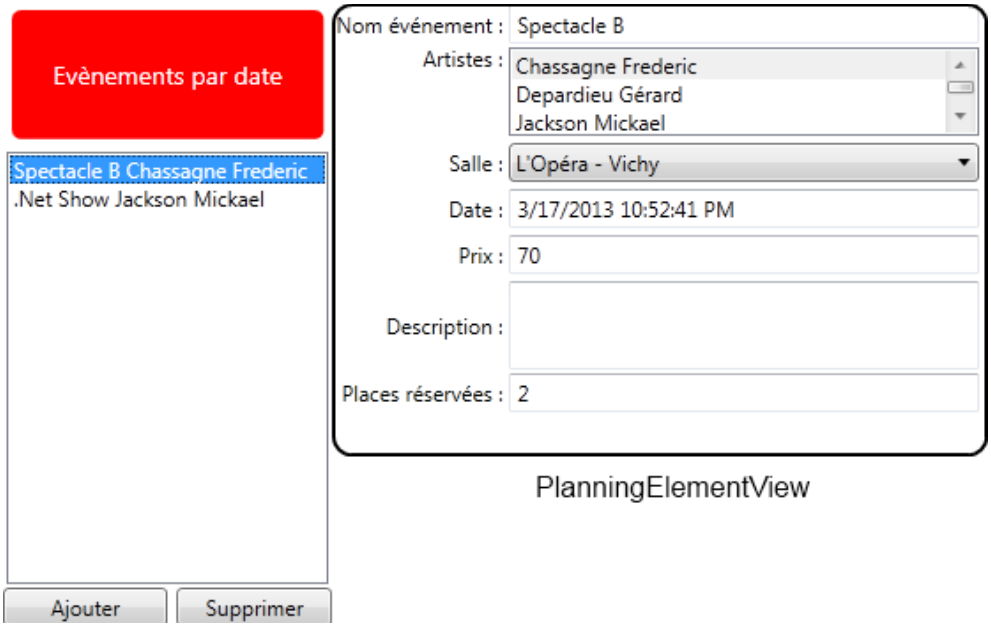
Remarque : Il s'agit de créer un « Planning Event ». Certains champs sont modifiables (Salle, Date et Nombre de places réservées), d'autres non. Par exemple, le prix est déterminé par le prix de l'évènement majoré de la commission de la salle. Il doit donc changer quand la salle change.

Etape 2 : M-V-VM : View

Ajouter un bouton à votre application WPF comme illustré ci-dessous :



Le clic sur la « gestion Evènements » devra ouvrir une nouvelle Window que vous nommerez GestionEvenementView et qui pourra ressembler au modèle ci-dessous :



PlanningElementView

Etape 3: M-V-VM : ViewModels

Toutes les classes ViewModel implémenteront la classe ViewModelBase. Penser à « binder » les ViewModel à leurs View, en utilisant le DataContext.

Créer la classe PlanningElementViewModel, ayant au moins un attribut de type PlanningElement. Cette classe exposera les propriétés à afficher dans la vue. Elle devra également notifier à la vue les modifications des propriétés exposées. (event OnPropertyChanged).

Créer la classe GestionEvenementViewModel, ayant au moins un attribut de type ObservableCollection<PlanningElementViewModel>. Enrichir cette classe afin qu'elle puisse gérer la liste des évènements du GestionEvementView.

A la fin de cette étape vous devez voir votre liste d'évènements par date s'afficher, lorsque vous cliquez sur un évènement, le détail s'affiche dans votre contrôle utilisateur.

Etape 4 : Les commandes

Plutôt que d'appeler directement des méthodes à partir des boutons (event_Click), les boutons seront bindés à des commandes.

La classe GestionEvenementViewModel possède des attributs de type ICommand pour chaque Command à gérer. Enrichir cette classe afin qu'elle gère l'ajout et la suppression de planningElement.

Conclusion

A la fin de ce TP, vous aurez une gestion complète des planningElement de votre Agenda. (Visualisation, Ajout, Suppression).