TP 12

Les objectifs de ce douzième TP sont :

- la conception d'interfaces graphiques en XAML,
- la gestion d'événements entre XAML et code-behind.

OBJECTIFS

Créer une application WPF correspondant à l'image de la page suivante. Ce magnifique croquis pourrait correspondre à un storyboard à utiliser comme spécifications d'interfaces qui aurait été écrit conjointement avec le client.

Puisqu'il s'agit de votre premier travail en WPF / XAML, voici quelques instructions pour vous guider.

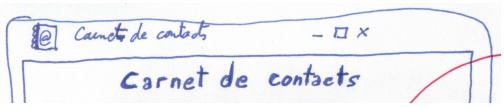
ORGANISATION

Créez une nouvelle solution qui contiendra deux projets :

- le projet giContacts qui vous est fourni (ENT),
- un projet de type «Application WPF», ayant dans ses références, le projet précédent.

MAIN WINDOW ET LAYOUT

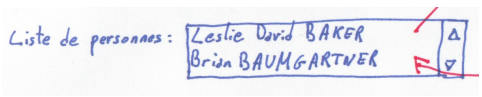
Réfléchissez au *layout* de votre fenêtre principale et proposez un brouillon. Modifiez le titre et l'icône de la fenêtre principale. Ajoutez un titre dans la fenêtre «Carnet de contacts».



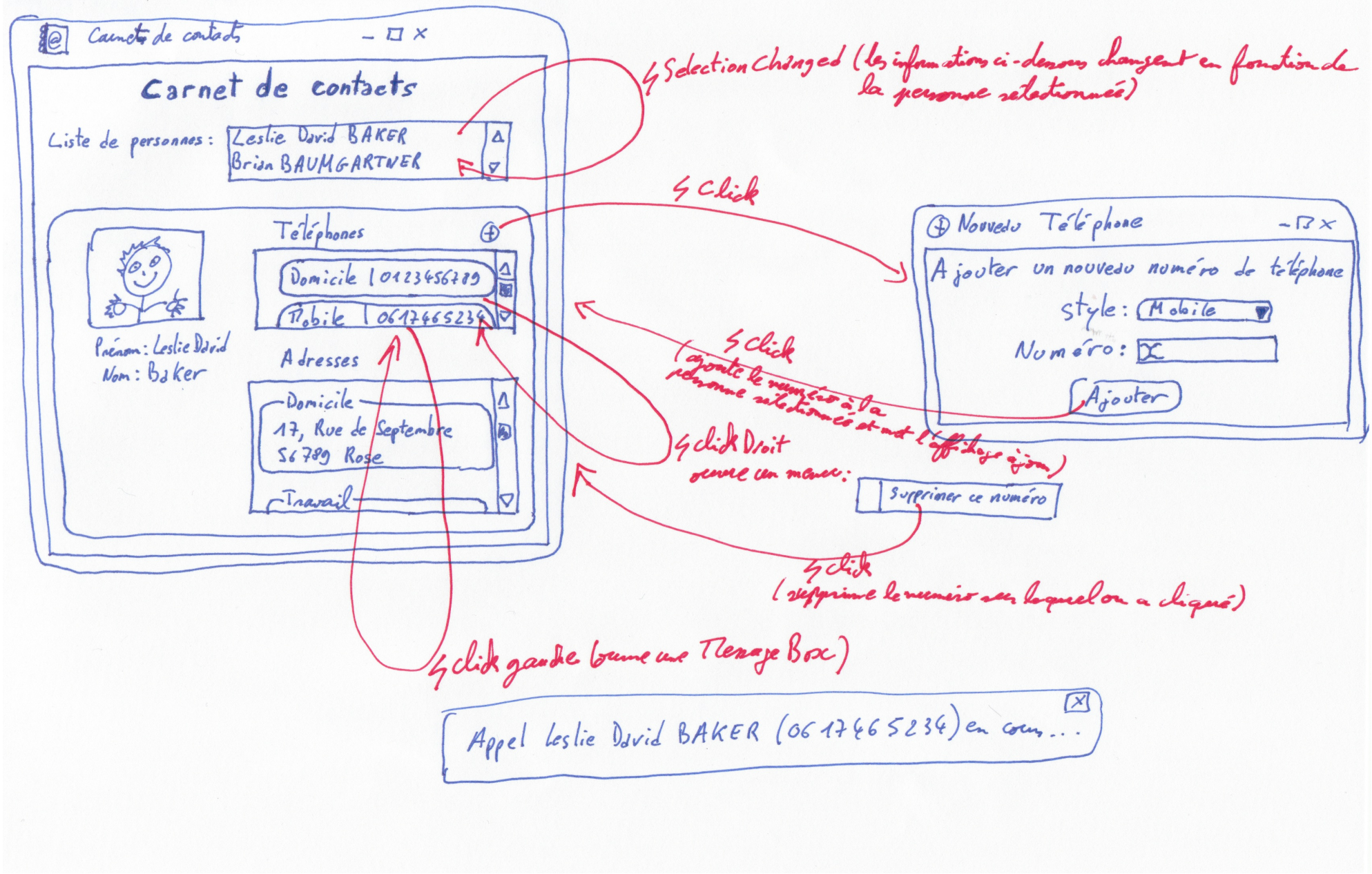
LISTBOX DES PERSONNES

Ajoutez la `ListBox` des personnes.

Pour tester l'apparence de cette `ListBox` en mode *design*, ajoutez 2 ou 3 personnes en XAML.

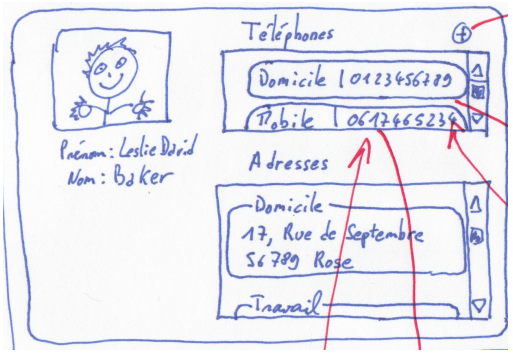


Ajoutez les personnes fournies dans la collection statique de la classe `giContacts.Personne` en *code-behind*. Constatez que l'affichage n'est pas convenable, et modifiez la méthode `ToString` de `Personne` pour obtenir le formatage demandé.


Carnet de contacts
Carnet de contacts
Liste de personnes :
Leslie David BAKER
Brian BAUMGARTNER
Téléphones
Domicile 0123456789
Mobile 0617465234
Adresses
Domicile
17, Rue de Septembre
56789 Rose
Travail
Prénom : Leslie David
Nom : Baker
Sélection Changed (les informations ci-dessous changent en fonction de la personne selectionnée)
Click
Nouveau Téléphone
Ajouter un nouveau numéro de téléphone
style : Mobile
Numéro :
Ajouter
Click (ajoute le numéro à la personne selectionné et met l'affichage ajour)
Click Droit ouvre un menu :
Supprimer ce numéro
Click (supprime le numéro sur lequel on a cliqué)
Click garde (sans une Message Box)
Appel Leslie David BAKER (0617465234) en cours...

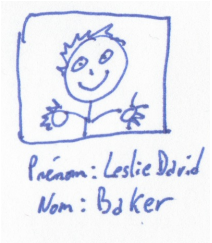
DÉTAILS D'UN CONTACT

Proposez un *layout* pour l'ensemble des détails d'un contact.



PHOTO, PRÉNOM ET NOM D'UN CONTACT

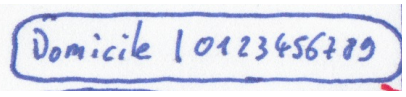
Codez la partie XAML permettant d'afficher la photo, le prénom et le nom d'un contact. Pour tester l'apparence en mode *design*, donnez des valeurs en XAML. Pour choisir une photo qui n'est pas dans l'assemblage, mais dans l'assemblage de giContacts, vous pouvez utiliser comme `Source = "/giContacts;Component/Images/mon_image.jpg"` (ceci sera vu lors du cours 13).



Gérez l'événement `SelectionChanged` sur la `ListBox` de personnes et modifiez le contenu de l'image, du prénom et du nom pour qu'ils correspondent à ceux de la personne sélectionnée dans la `ListBox`.

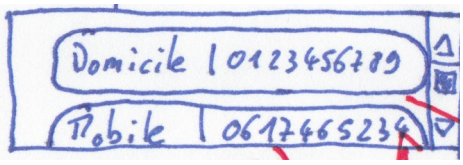
TÉLÉPHONE USERCONTROL

Créez un `UserControl` pour représenter un téléphone. Ce `UserControl` contiendra donc une propriété de type `giContacts.Téléphone`. Ce `UserControl` affichera le type du téléphone et le numéro de téléphone. Donnez une valeur par défaut en XAML pour tester l'apparence en mode *design*, mais faites en sorte, en *code-behind*, que l'appel du setter de la propriété de type `giContacts.Téléphone` modifie les contrôles du `UserControl` pour afficher les informations relatives au téléphone choisi.



LISTE DE TÉLÉPHONES

Dans la fenêtre principale, ajoutez une `ListView` qui contiendra des numéros de téléphone (i.e. le `UserControl`). Testez l'apparence en mode *design* en ajoutant en XAML deux numéros de téléphone provenant des ressources statiques. Pour cela, ajoutez sous la balise `Window`, les lignes suivantes :



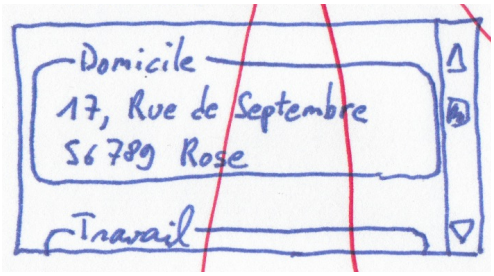

```
<Window.Resources xmlns:contacts="clr-namespace:giContacts;assembly=giContacts">
        <contacts:Téléphone x:Key="tel1" Type="Domicile"
Numéro="0123456789"/>
        <contacts:Téléphone x:Key="tel2" Type="Mobile" Numéro="0678901234"/>
 </Window.Resources>
```

Pour utiliser un de ces `giContacts.Téléphone`, vous pouvez utiliser `{StaticResource tel1}`. Ceci sera montré et expliqué pendant le cours 13.

Modifiez le code-*behind* pour qu'à l'exécution, la `ListView` de `Téléphone` contiennent les téléphones de la personne sélectionné dans la `ListBox`.

LISTE D'ADRESSES

Recommencez les deux étapes précédentes mais cette fois-ci pour les adresses.



MESSAGEBOX POUR L'APPEL EN COURS DEPUIS LE USERCONTROL

Modifiez le `UserControl` du téléphone pour qu'un clic sur un téléphone ouvre une `MessageBox` indiquant «*Appel en cours…*».

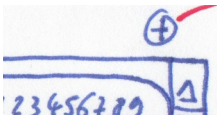
MESSAGEBOX POUR L'APPEL EN COURS AVEC ROUTED EVENT

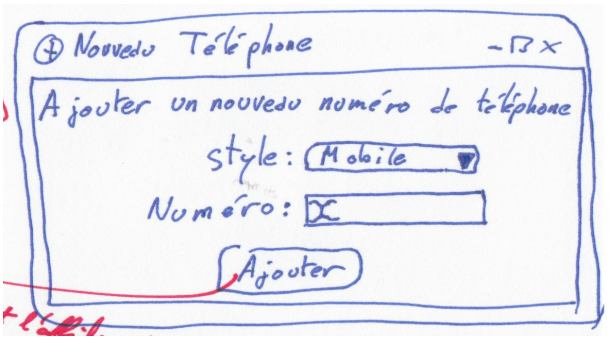
Sans modifier le `UserControl` du téléphone, faites maintenant en sorte qu'un clic sur un numéro de téléphone lance une `MessageBox` indiquant «*Appel Prénom NOM (numéro) en*

cours...». Notez que le `UserControl` ne sait pas qui est le propriétaire du numéro. Vous allez donc être obligé de jouer avec les *RoutedEvents*. Captez l'événement depuis la fenêtre principale mais identifiez quelle est la source. En fonction de cette source et de la personne sélectionnée, vous devriez pouvoir afficher les informations demandées.

AJOUT D'UN NOUVEAU NUMÉRO DE TÉLÉPHONE

Ajoutez un bouton qui, lorsqu'on le clique, ouvre une nouvelle fenêtre. Cette nouvelle fenêtre permettra de choisir le type de numéro de téléphone, le numéro et lors du clic sur ajouter, fermera la fenêtre et ajoutera le numéro à la liste des numéros de la personne sélectionnée. Vous pourrez notamment pour cela, utiliser l'événement `Closing` de cette nouvelle fenêtre qui est lancé juste avant sa fermeture. Faites également en sorte que cette fenêtre bloque l'accès à la fenêtre principale (c'est-à-dire tant qu'elle n'est pas fermée).





SUPPRESSION D'UN NUMÉRO DE TÉLÉPHONE

Gérez le clic droit sur les numéros de téléphone pour qu'un menu contextuel s'ouvre et propose *«retirer ce numéro de téléphone»*. Gérez le clic sur cet item de menu pour que le numéro sélectionné soit effectivement retiré.