

## Description du thème

Propriétés	Description
<b>Intitulé long</b>	Cinquième TP d'une série permettant la découverte d'une application mobile sous les systèmes d'exploitation Windows Phone 7, 7.5 ou 8
<b>Formation concernée</b>	BTS Services informatiques aux organisations
<b>Matière</b>	SLAM 4
<b>Présentation</b>	Ces TP proposent de développer une application mobile à différentes itérations du cycle de développement
<b>Notions</b>	<p><u>Savoirs</u></p> <ul style="list-style-type: none"> <li>• D4.1 - Conception et réalisation d'une solution applicative</li> <li>• D4.2 - Maintenance d'une solution applicative</li> </ul> <p><u>Savoir-faire</u></p> <ul style="list-style-type: none"> <li>• Programmer un composant logiciel</li> <li>• Exploiter une bibliothèque de composants</li> <li>• Adapter un composant logiciel</li> <li>• Valider et documenter un composant logiciel</li> <li>• Programmer au sein d'un framework</li> </ul>
<b>Pré-requis</b>	Développement objet, C #, Visual-Studio
<b>Outils</b>	Un environnement de développement pour les applications mobiles Windows Phone. Visual Studio 2010 express pour application Windows Phone au minimum ou Visual Studio 2010 et le kit de développement Windows Phone.
<b>Mots-clés</b>	Application mobile, C#
<b>Durée</b>	2 h
<b>Auteur(es)</b>	Patrice Grand
<b>Version</b>	v 1.0
<b>Date de publication</b>	Mars 2015

## Énoncé

La suite du développement est consacrée à la visualisation et la suppression des offres du chauffeur connecté. Ceci doit être géré à partir du menu initial « mes offres ». Une partie du code est déjà écrit par votre collègue ; vous êtes chargé de terminer cette partie.

Votre collègue vous informe de l'état du développement :

<<

- *Votre collègue : Je me suis concentré sur la seule partie concernant les offres dont le départ est le domicile. Dans la partie modèle, j'ai écrit la fonction qui retourne la liste des offres du chauffeur connecté. Côté XAML, je pense que tu n'as rien à toucher ; j'ai créé un template dans app.xaml et bindé les éléments aux champs :*

```

<TextBlock Text="{Binding jour}" FontSize
<TextBlock Text="{Binding heure}" Height="
<TextBlock Text="{Binding date}" Height="ε
<TextBlock Text="{Binding lieu}" Height="ε

```

Mon petit problème c'est que ça ne marche pas !!

- Vous : Et qu'est ce qui ne marche pas ?
- Votre collègue : Ca ne supprime pas l'offre sélectionnée !! Le code est pourtant simple, regarde :

```

private void lstDepartEntreprise_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    modele.OffreDepartEntreprise offre ;
    offre = (modele.OffreDepartEntreprise)e.AddedItems[0];
    var confirmer = MessageBox.Show("Confirmez la suppression", "Supprimer une offre", MessageBoxButton.OKCancel);
    if (confirmer == MessageBoxResult.OK)
    {
        this.mesOffresDepartEntreprise.Remove(offre);
        // this.lstDepartEntreprise.Items.Remove(offre); code qui plante
    }
}

```

Lorsque l'on sélectionne une offre, on récupère l'offre, on demande la confirmation, un détail ici, et l'on supprime l'offre de la liste. En fait, elle est bien supprimée de la liste mais l'affichage continue à montrer cette offre !! Comme si la vue n'était pas « informée » du changement de la liste.

- Vous : Oui, je vois ; et la ligne que tu as laissée en commentaire ?
- Votre collègue : C'est un autre essai ; à la place de supprimer dans le modèle, j'ai supprimé dans la listBox. Mais là c'est pire ; ça plante !! j'ai une erreur, regarde :

Exception snapshot:

System.InvalidOperationException	{"Operation not supported on read-only collection."}
[System.InvalidOperationException]	{"Operation not supported on read-only collection."}
Data	{System.Collections.ListDictionaryInternal}
Message	Operation not supported on read-only collection.
StackTrace	at System.Windows.Controls.ItemCollection.RemoveImpl(Ob

- Vous : Et tu as une idée ?
- Votre collègue : J'ai vu sur internet que l'on ne pouvait binder une liste à une listBox que dans le cas où on ne modifie pas la liste (comme pour le formulaire des offres disponibles). Dans le cas contraire, il faut utiliser une classe ObservableCollection.
- Vous : Il faut modifier le modèle pour retourner un tel objet plutôt qu'une liste?
- Votre collègue : Non, tu peux le faire dans le formulaire et créer ton objet ObservableCollection à partir de ta liste.
- Vous : Ok, je vais voir.

### Travail à faire.

- Terminer la gestion des suppressions des offres dont le départ est l'entreprise en tenant compte des indications fournies par votre collègue. Temps estimé : 1 heure 30.
- Créer le formulaire permettant la suppression des offres dont le départ est le domicile. Temps estimé : 30 minutes.