

Description du thème

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

Énoncé

L'application sur laquelle vous travaillez a avancée : le formulaire de présentation des offres de covoiturage dont le départ est l'entreprise a été réalisé. Ce travail a été accompli par un collègue de l'équipe un peu plus familier avec cette technologie ; vous prenez connaissance du code.

Plusieurs points vous semblent un peu obscurs, vous demandez à votre collègue un peu plus aguerri qu'il vous donne quelques explications complémentaires :

<<

- *Votre collègue : Le formulaire de présentation des offres utilise le mode « panorama », la même page Xaml est décomposée en sous-pages pour l'utilisateur, qui peut ainsi naviguer en balayant de droite à gauche. L'utilisateur a l'impression de faire défiler plusieurs pages mais en fait, ce n'est qu'une partie d'une même page qui est proposée par l'intermédiaire de la collection Controls : Panorama et des Panoramaltems. Un autre mécanisme est disponible : le Controls :Pivot, qui fait défiler les pages de haut en bas.*
- *Vous : Il faut écrire tout cela en Xaml.*

- Votre collègue : Moi j'utilise Visual Studio qui permet d'ajouter une page de « type panorama » et qui fournit le squelette.
- Vous : Je préfère cela... Je vois des expressions entre accolades dans le code Xaml.
- Votre collègue : Oui lorsque tu veux utiliser des ressources existantes tu mets entre accolades ton appel ; c'est un peu comme les styles CSS. Tu peux aussi créer tes propres ressources dans un fichier Xaml ou dans la page où tu l'utilises.
- Vous : Tu utilises le Binding de données...
- Votre collègue : Oui c'est assez classique, tu définis ton contexte de Binding (DataContext) que tu associes à ton contrôle ou ta page et tu bind ensuite sur une propriété que tu lies à une donnée en utilisant à nouveau des accolades. Le contexte de Binding peut être une liste ou un simple objet !!
- Vous : Ok...>>

Travail à faire

Dans quel fichier se trouve la ressource créée évoquée par le collègue ?

Que fait cette ressource ?

Comment est appelée cette ressource au moment de son utilisation ?

Tâches suivantes

On vous confie la responsabilité de créer la page de présentation d'une offre ; elle devra se présenter ainsi :



Lorsque l'utilisateur tapera sur les images, ceci devra lancer les applications d'envoi de mail ou de téléphonie du téléphone. Les images sont disponibles dans le répertoire *images*.

Avant de vous lancer, vous repérez que l'appel de la page d'une offre est écrit :

```
private void lst_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    modele.OffreDepartEntreprise offre = (modele.OffreDepartEntreprise)e.AddedItems[0];
    PhoneApplicationService.Current.State["chauffeur"] = offre.chauffeur;
    NavigationService.Navigate(new Uri("/uneOffre.xaml", UriKind.Relative));
}
```

Vous êtes un peu intrigué et vous demandez l'aide de votre collègue :

<<

- Votre collègue : la première ligne permet de récupérer l'offre sélectionnée ; on demande à l'objet source du clic (l'objet e) de donner l'offre concernée contenue dans la listBox comme premier item –puisque l'on ne sélectionne qu'une offre- (AddedItems[0]).
- Votre collègue : à la deuxième ligne on crée dans un dictionnaire (dictionary) disponible au niveau de toute l'application une nouvelle « entrée » que l'on nomme chauffeur. Ce dictionnaire est disponible dans les autres pages.
- Vous : Ok>>

Travail à faire

Réaliser le formulaire demandé. Vous devrez utiliser le Binding pour afficher les données.

Temps estimé 2 heures.