Plan de rencontre:

**MODÈLE**

1. Créer les classes du modèle des données (une classe par table projetée)
2. Entity framework
   1. Inclure la librairie Entity framework via (Manage Nuget Packages Manager)
   2. Ajouter la chaine de connexion à la BD dans Web.config
3. Créer la classe d’interface à la BD dérivant de DbContext
4. Créer les extensions pour constituer le DAL (ensembles de méthodes pour les opérations CRUD sur les tables de la BD)
5. Une fois la BD générée, inclure le fichier \*.mdf dans le répertoire App\_Data
6. Créer les classes spécialisées aux vues des contrôleurs l’application
   1. Attributs de validation et d’affichage
7. Gestion des images (externe à la base de données) Voir la classe ImageGUIDReference
   1. Ajouter un membre dans la classe spécialisé aux vues

public ImageGUIDReference PhotoReference { get; set; }

* 1. Dans le constructeur initialiser le répertoire des images et l’url de l’image par défaut
  2. Ajouter au projet le répertoire des images
  3. Créer les méthodes CRUD des images
     1. GetPhotoURL()
     2. UpLoadPhoto(HttpRequestBase Request)
     3. RemovePhoto()

1. Une fois la BD générée, inclure le fichier \*.mdf dans le répertoire App\_Data
2. Bâtir les actions CRUD du contrôleur
3. Enctype pour les formulaires comportant un contrôle d’upload de fichier

@using (Html.BeginForm("Create", "Contacts", FormMethod.Post, new { enctype = "multipart/form-data" }))

…

@Helper.ImageUpLoader(Url, Model.GetPhotoURL(), "Cliquez pour choisir la photo...", "max-width:280px;")

1. Les html helpers (App\_Code, Extensions, View/Web.config))
2. Javascript (mask, calender, etc.) et bundle….