

Mini-Projet de .NET

Description

L'application qu'il faudra réaliser dans ce projet est un « Picasa-like ». Le but de cette application est de proposer à l'utilisateur un outil lui permettant de visualiser ses photos personnelles. Ce même outil devra aussi lui permettre de créer des albums en ligne à partir de ses photos en local et de les partager avec d'autres utilisateurs. Une interface web permettra à chaque utilisateur de voir ses albums ainsi que ceux qu'on lui a partagés sous la forme d'un diaporama.

Architecture

L'application sera basée sur une architecture 3-tiers (Figure 1) :

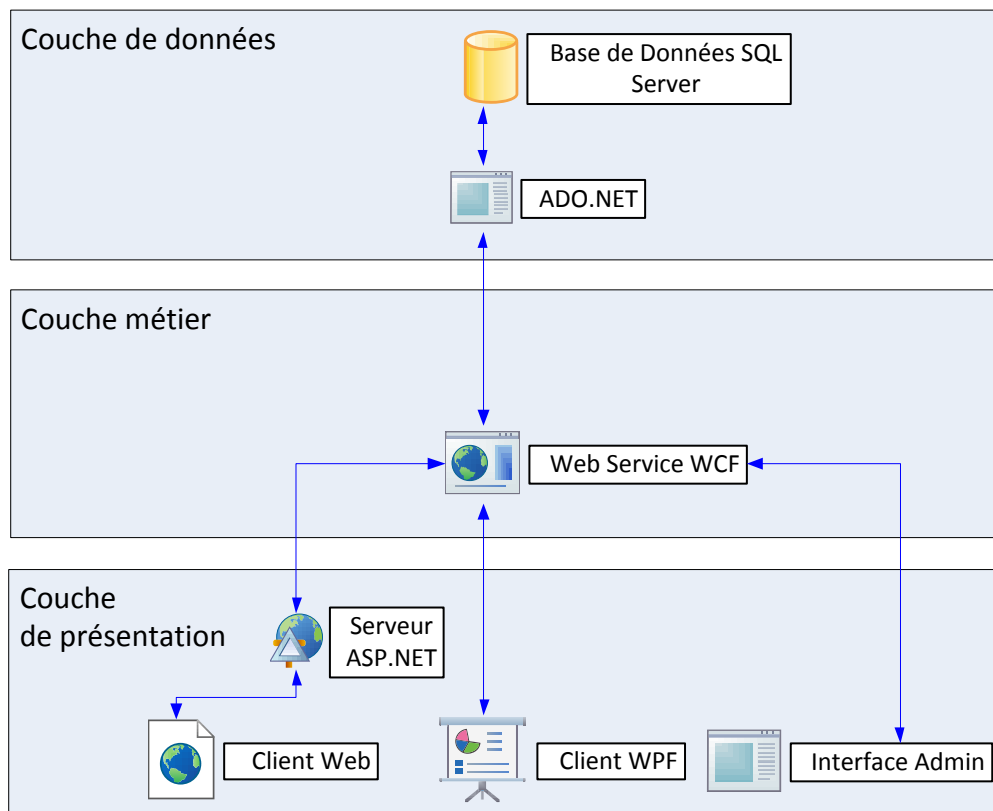


Figure 1 : Architecture Logicielle de l'application

Couche Présentation

3 Clients seront demandés :

Une interface administrateur

Ce client devra permettre à l'administrateur système de supprimer n'importe quel compte utilisateur, album photo ou photo. L'implémentation pourra se faire en mode console.

Le client lourd WPF

Le client lourd WPF permettra à l'utilisateur de visualiser ses photos et de créer des albums en ligne. La structure de l'interface ainsi que son design n'est pas imposé. Néanmoins, afin de mieux illustrer les différentes fonctionnalités qui sont demandées, le schéma d'un exemple d'interface graphique est donné en Figure 2.

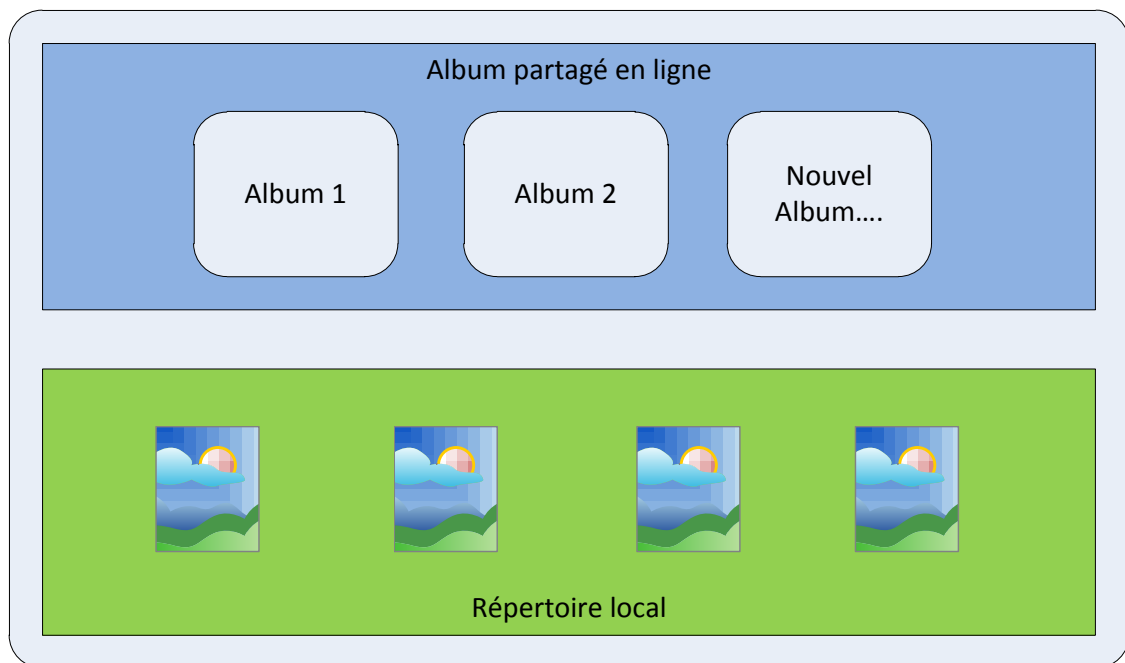


Figure 2 : Schéma d'interface graphique

La première zone (en vert) permet à l'utilisateur de sélectionner un répertoire en local et de visualiser les photos présentes dans ce répertoire. La seconde zone (en bleu) permet de gérer les albums en ligne. Cette zone affiche la liste des albums en ligne appartenant à l'utilisateur. L'utilisateur peut alors créer un nouvel album ou visualiser un album existant.

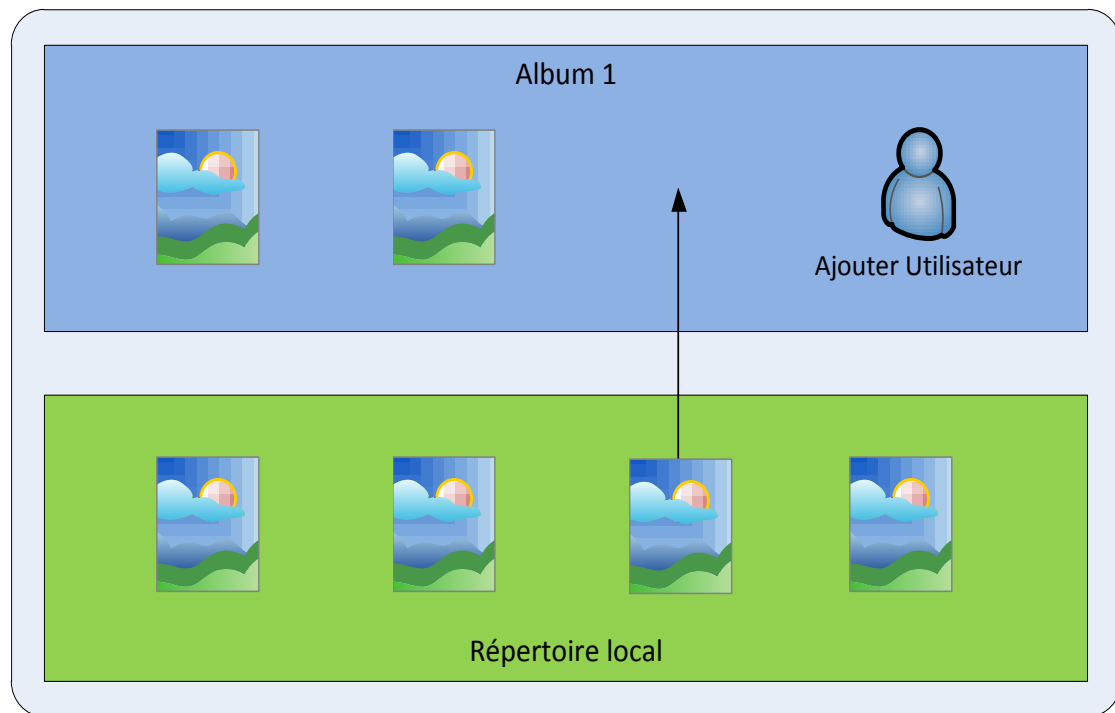


Figure 3 : Drag and drop de la photo d'un espace à un autre

Si l'utilisateur sélectionne un album existant (Figure 3), les photos seront téléchargées une à une à partir du serveur et pourront être visualisées dans l'interface (zone bleu). L'utilisateur pourra ainsi faire un drag and drop pour copier la photo sur son disque dur. Dans l'autre sens, l'utilisateur pourra faire un drag and drop d'une photo présente sur son disque dur dans un album en ligne afin que celle soit téléchargée sur le serveur, comme sur l'exemple de la Figure 3. Une option permettra d'autoriser un autre utilisateur à visualiser cet album. Cet album apparaîtra donc dans la liste d'albums partagés sur le client web de cet autre utilisateur.

Le client léger ASP.NET

Ce client permettra à l'utilisateur de voir sous la forme de diaporamas les différentes photos de ces albums en ligne ainsi que ceux que lui ont partagés les autres utilisateurs. Il n'est demandé aucune fonctionnalité autre que la visualisation des photos pour ce client.

Couche de traitement

Cette couche devra exposer les services suivants aux différents clients :

```
interface IService
{
    String addUser(String user, String password);
    String createNewAlbum(String wantedName, String user);
    void deleteAlbum(String albumID);
    List<String> getUserAlbum(String user);
    List<String> getViewableAlbum(String user);
    void allowUserAlbum(String albumID, String user);
    void disallowUserAlbum(String albumID, String user);
    String addPicture(String albumID, Stream picture);
    void deletePicture(String pictureID);
}
```

Tous les utilisateurs n'auront pas accès à l'ensemble des services. Cette liste de services n'est pas exhaustive et vous pouvez l'enrichir ou la modifier si vous en jugez le besoin. Ces services seront implémentés en WCF. Il est conseillé d'utiliser l'encodage des messages en Mtom pour les services qui géreront le transfert des fichiers. Vous devez par ailleurs implémenter un mécanisme d'authentification et d'autorisations par les rôles pour ces différents services. Des précisions concernant cette gestion de la sécurité vous seront apportées dans un énoncé complémentaire.

Couche d'accès aux données

Cette couche contiendra les couches d'accès aux données. Il est conseillé d'utiliser une base de données SQL Server 2008 avec ADO.NET (mode déconnecté ou connecté). Une autre technologie peut toutefois être acceptée si vous justifiez son utilisation dans le rapport. Cette couche de donnée présentera une interface définissant les différentes méthodes d'accès aux données. Vous devrez aussi implémenter un mécanisme de transactions pour les opérations qui le nécessitent.

Planning

Ce projet se déroulera pendant les heures de TD ainsi qu'à la maison si nécessaire. Le projet se découpera en 4 séances de 2 heures encadrées. A chaque séance, un énoncé complémentaire vous sera apporté afin de vous aider à réaliser les différentes étapes du projet. Une dernière séance libre vous permettra de finir d'implémenter les différentes fonctionnalités de votre projet. Le rendu devra se faire au plus tard 2 semaines après la fin du dernier TD.

Séance 1 :

- Design de la base de données
- Création de la base de données
- Définition de l'architecture et des interfaces

Séance 2 :

- Implémentation des méthodes d'accès aux données
- Création de l'interface admin
- Implémentation du web Service

Séance 3 :

- Création de l'interface Web ASP.NET
- Implémentation des transactions

Séance 4 :

- Intégration de la sécurité
- Création de l'interface client lourd en WPF

Séance 5 :

- Séance libre

Modalités de rendu

Ce projet doit être réalisé en groupe de 2 personnes. Il devra être rendu au plus tard 2 semaines après la dernière séance TD sous la forme d'une archive dans l'entrepôt Polytech'. Cette archive devra contenir l'ensemble des projets nécessaires pour faire fonctionner l'application ainsi qu'un mini-rapport de quelques pages. Ce rapport devra décrire l'architecture et les fonctionnalités de votre projet. Il est conseillé pour cela d'y inclure un schéma de type FMC des différents composants de votre architecture, un diagramme entité-association représentant votre base de données et des diagrammes de séquence représentant quelques cas d'utilisation. Enfin ce rapport devra comporter un paragraphe de prise de recul. Dans cette partie, vous devrez expliquer les différentes améliorations que vous auriez pu apporter à votre projet si vous aviez eu plus de temps. Si vous regrettez certains choix technologiques, vous devrez aussi expliquer pourquoi vous les avez faits, pourquoi ils ne se sont pas révélés optimaux et quelles alternatives vous adopteriez si vous aviez à refaire le projet.

Fonctionnalités minimales demandées

Voici résumée la liste des fonctionnalités que devront permettre vos différents clients :

Client console :

- Création/Suppression d'utilisateurs
- Suppression d'un album
- Suppression d'une photo

Client lourd :

- Création d'un album
- Visualisation des photos d'un répertoire local
- Visualisation des photos d'un album distant
- Download/Upload de photos d'un album distant

Client léger :

- Visualisation de la liste des albums partagés
- Visualisation des photos d'un album partagé

Bonus

Toute fonctionnalité supplémentaire, (exemple : réalisation d'un client web en Silverlight, d'une interface pour WP7, application de filtres sur les photos pour le client WPF, formulaire ASP.NET pour créer de nouveaux utilisateurs, etc...) sera la bienvenue et prise en compte en bonus dans la note finale.