

Application de gestion de tournois sportifs

Objectifs :

- ✓ Utiliser des classes *métier*
- ✓ Utiliser l'outil WPF
- ✓ Utiliser les paramètres d'application
- ✓ Utiliser le binding
- ✓ Développer un module d'accès à une base de données via LINQ

I. Présentation du TP :

Il s'agit de réaliser une application Windows qui permet la gestion de tournois dans un complexe multisports.

II. Expression du besoin :

L'application à développer devra être constituée de deux parties principales :

- ✓ Une première partie "Gestion d'un tournoi", qui permettra d'ajouter, modifier ou supprimer les informations relatives aux tournois, aux sports, aux participants et aux gestionnaires.

L'accès à cette partie doit logiquement être restreint au gestionnaire du complexe de sports. Cet accès sera protégé par un login et un mot de passe.

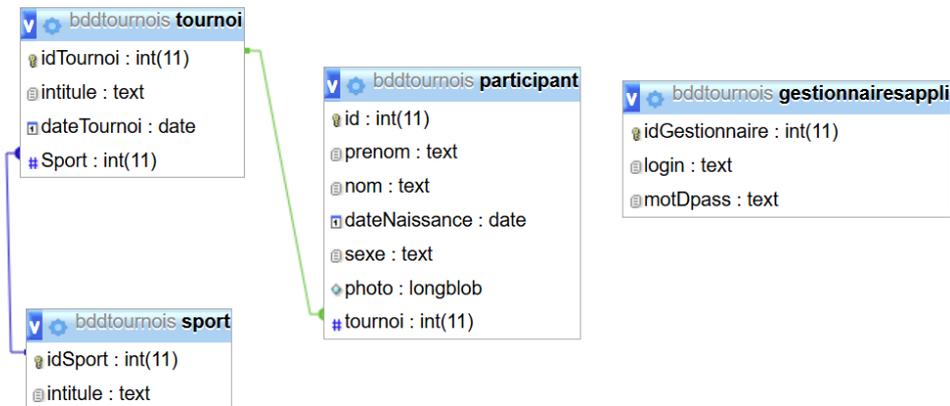
- ✓ Une seconde partie "Consultation", qui permettra de consulter les informations sur les tournois, les sports ou les participants, de faire différentes recherches.

Cette partie sera accessible à tous et ne nécessitera donc pas de s'identifier. Les différentes consultations possibles de la partie "Consultation" seront les suivantes :

- Afficher toutes les informations sur tous les participants (prénom, nom, date de naissance, sport, photo d'identité. L'utilisateur aura la possibilité de faire un tri par ordre alphabétique des noms des participants.
- Effectuer une recherche d'un participant par son nom en spécifiant tout ou partie du nom recherché.
- Effectuer une recherche des participants par tournoi en sélectionnant un tournoi parmi les tournois enregistrés
- Effectuer une recherche des tournois par sport en sélectionnant un sport parmi les sports enregistrés.

Dans tous les cas le résultat de la recherche sera affiché sous forme de tableau ou de liste.

L'ensemble des données des tournois organisés est stocké dans une base de données MySQL ***bddTournois***, dont le schéma est présenté ci-dessous.

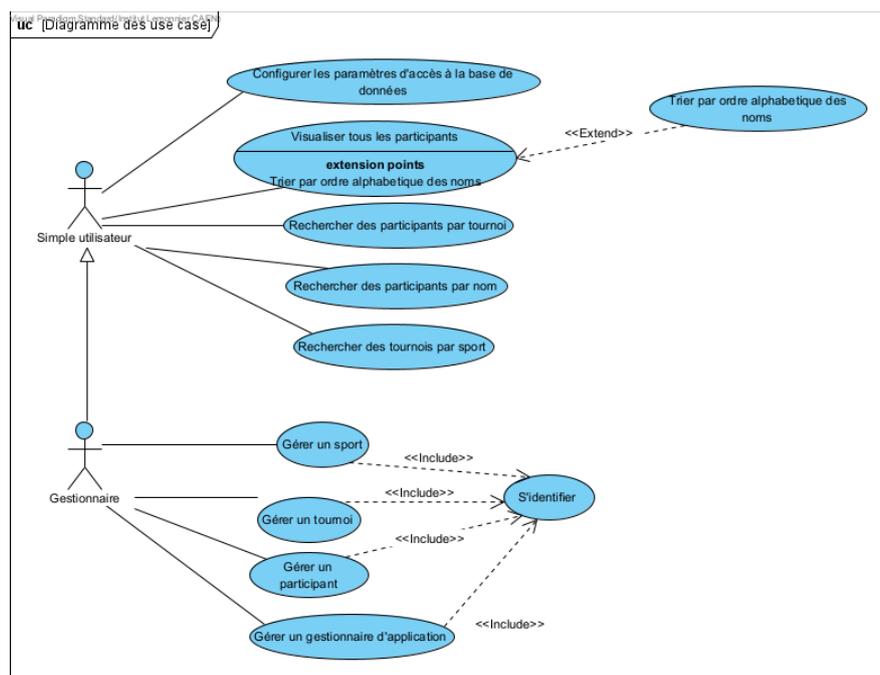


Cette base de données est constituée de quatre tables :

- La table "*gestionnaireappli*" qui contient les logins et les mots de passes des administrateurs de l'application.
- La table "*sport*" qui contient les différents sports proposés par le complexe.
- La table "*tournoi*" qui contient les différents tournois pouvant être organisés dans le complexe sportif.
- La table "*participant*" qui contient les différentes informations associées aux différents participants aux tournois organisés.

L'application à développer met en évidence deux acteurs, l'acteur "**Gestionnaire**" autorisé à modifier le contenu de la base de données après authentification et l'acteur "**Simple utilisateur**" qui est uniquement autorisé à la consultation des données.

Ces deux acteurs interviennent dans les scénarios suivants :



Cas d'utilisation de l'acteur Gestionnaire :

Cas d'utilisation "S'identifier" :

Description : le gestionnaire doit être préalablement authentifié afin de pouvoir accéder à la gestion des participants, des tournois, des sports et des gestionnaires.

Une fois l'authentification validée les menus ou boutons permettant l'accès à la gestion (tournois, sports et participants) seront dégrisés.

Cas d'utilisation "Gérer un sport" :

Description : le gestionnaire peut consulter les sports existants, sélectionner un sport pour le modifier ou le supprimer ou bien ajouter un nouveau sport à la liste.

Cas d'utilisation "Gérer un tournoi" :

Description : le gestionnaire peut consulter les tournois existants, sélectionner un tournoi pour le modifier ou le supprimer ou bien ajouter un nouveau tournoi à la liste.

Cas d'utilisation "Gérer un participant" :

Description : le gestionnaire peut consulter les participants existants, sélectionner un participant pour le modifier ou le supprimer ou bien ajouter un nouveau participant à la liste.

Cas d'utilisation "Gérer un gestionnaire d'application" :

Description : le gestionnaire peut consulter les gestionnaires existants, sélectionner un gestionnaire pour le modifier ou le supprimer ou bien ajouter un nouveau gestionnaire.

Cas d'utilisation de l'acteur Simple utilisateur :

Cas d'utilisation " Configurer les paramètres d'accès à la base de données" :

Description : le gestionnaire et/ou le simple utilisateur configure les paramètres d'accès à la base de données grâce à une fenêtre dédiée et accessible depuis un bouton de menu de l'application. Les paramètres sont :

- ✓ L'adresse IPv4 du serveur de base de données,
- ✓ Le port d'écoute du service base de données,
- ✓ Le login du compte d'accès à la base de données,
- ✓ Le mot de passe du compte d'accès à la base de données,
- ✓ Le nom de la base de données.

Cas d'utilisation " Visualiser tous les participants" :

Description : le simple utilisateur demande l'affichage de tous les participants avec l'ensemble des informations disponibles. Il peut s'il le souhaite effectuer un tri par ordre alphabétique des noms des participants.

Cas d'utilisation " Rechercher un participant par nom" :

Description : La recherche impose de taper tout ou partie du nom d'un participant dans une zone d'édition prévue à cet effet. À la suite de cela le résultat de la

recherche s'affiche sous forme d'un tableau de participants, ayant ce nom, avec l'ensemble des informations disponibles.

Cas d'utilisation " Rechercher des participants par tournoi" :

Description : La recherche impose au préalable de sélectionner un tournoi depuis la liste de l'ensemble des tournois. À la suite de cette sélection le résultat de la recherche s'affiche sous forme d'un tableau de participants. En sélectionnant un participant, on peut afficher toutes ses informations.

Cas d'utilisation " Rechercher des tournois par sport" :

Description : La recherche impose au préalable de sélectionner un sport depuis une liste de l'ensemble des sports. À la suite de cette sélection le résultat de la recherche s'affiche sous forme d'une liste de tournois avec l'ensemble des informations disponibles.

III. Contraintes de réalisation :

I. Contraintes de réalisation

Langages : C#, XAML, LINQ.

Ressources logicielles : Visual Studio 2022, LINQconnect Express (Devart Software).

Ressources pédagogiques : Cours et TDs C#, WPF,

II. Spécifications du serveur de bases de données

Adresses IP, port d'écoute du service MySQL et comptes d'accès :

- ✓ Privée : **xxx.xxx.xxx.xxx** (à voir en TP)
- ✓ Port d'écoute TCP : **3306**,
- ✓ Nom de la base de données : **bddtournois**,
- ✓ Compte d'accès :
 - Login : **adminTournois**,
 - Mot de passe : **Password1234@**

IV. Réalisation :

1. Installer la base de données MySql (script fourni)
2. Créer une solution vide et la nommer "*SolutionTournoi*",
3. Ajouter à la solution "*SolutionTournoi*" un projet de type "**Application WPF .Net Framework**" et le nommer "*AppTournoi*",
4. Ajouter à la solution "*SolutionTournoi*" un projet de type bibliothèque de classe (.Net Framework) et le nommer "*DllTournois*". *Renommez Classe1.cs en bddtournoi.cs*. Cette dll utilisera les classes créées automatiquement par l'outil **Linq**, pour accéder à la base de données.

5. Installation et utilisation de l'outil Linq ;

Si l'installation est à faire, la partie suivante est pour vous :

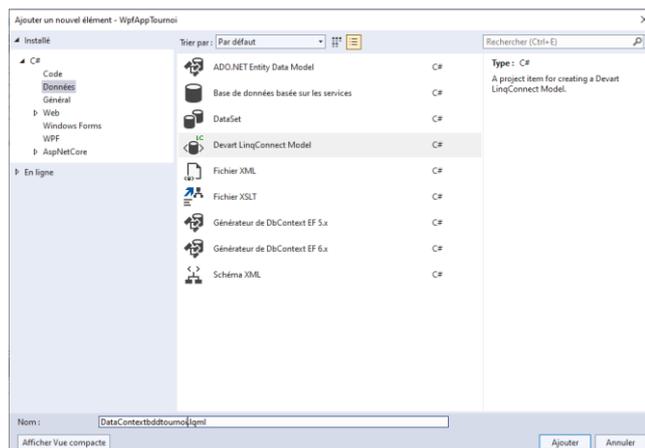
<https://www.devart.com/linqconnect/>

La version Linq Connect express est gratuite et prend en charge .Net Framework. Cette version est limitée à 10 entités c'est à dire 10 tables dans la base de données. (L'installation doit être faite avec Visual studio fermé. Choisir l'installation par défaut qui installe tous les outils nécessaires ainsi qu'une aide à l'utilisation de Linq).

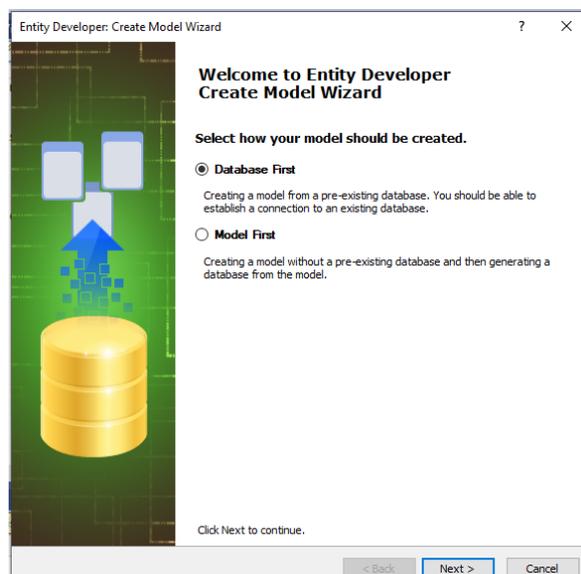
Une fois Linq installé :

Pour installer "la relation" entre la base de données *bddtournois* et *DllTournoi* il faut procéder comme suit :

- Faire un clic droit sur le projet "*DllTournois*" → ajouter → nouvel element : choisir le modèle *DevartLinqConnectModel*, changer le nom en *DataContextBddTournoi.lqml* puis ajouter,



→ Choisir Choisir DataBase first puisque la base bddtournois est déjà créée.



→ Faire un clic sur Next et remplir les champs selon les paramètres du serveur et de la base de données.

Entity Developer: Create Model Wizard

Set up data connection properties
On this page, specify connection settings for your model.

Provider: Devart dotConnect for MySQL

Connection Parameters

Host: NZA-PC

Port: 3306

User Id: adminTournoi

Password: *****

Allow saving password

Database: bddtournoi

Connection String
user id=adminTournoi;password=*****;host=NZA-PC;database=bddtournoi;persist security info:

Test Connection Advanced...

< Back Next > Cancel

→ Cliquez sur Next,

Entity Developer: Create Model Wizard

Choose Model Contents
On this page you can choose model content.

What should the model contain?

Generate From Database
Generates the model from a database. This wizard also lets you specify the database objects to include in the model.

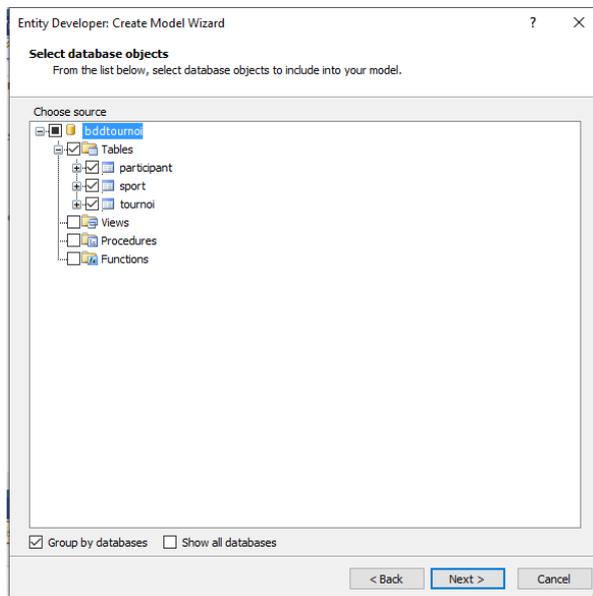
Empty Model
Creates an empty model as a starting point for designing a model visually. You can add database objects to the model later by the help of Database Explorer or Update From Database Wizard.

< Back Next > Cancel

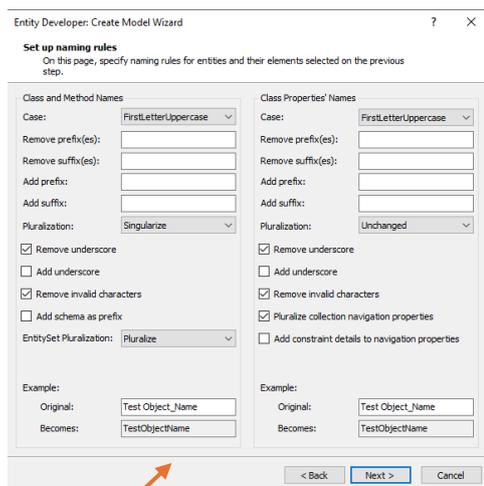
→ Choisir le modèle généré depuis la base de données

→ Un clic sur Next permet à Linq de rechercher dans la base *bddtournois* les tables et permet de sélectionner celles qu'on souhaite intégrer au projet. On souhaite

intégrer toutes les tables de la base de données, il faut donc laisser la sélection telle quelle et poursuivre.

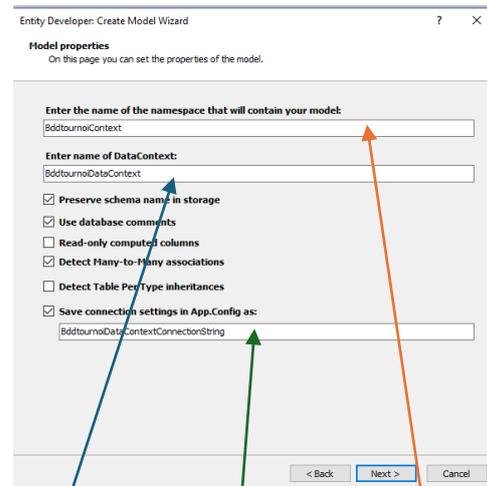


→ Cliquez sur Next 2 fois



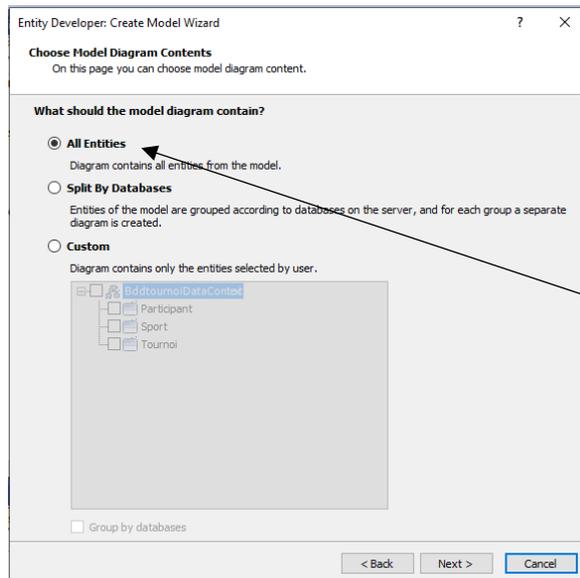
Sur cette fenêtre il est possible de modifier des noms, changer la casse etc.

Pour conserver les mêmes noms des tables, ne rien modifier et poursuivre.



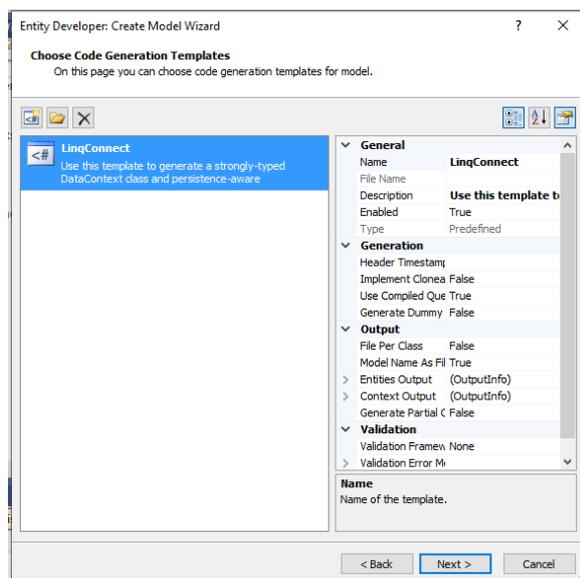
Linq crée un namespace **BddtournoiContextet**,
Crée dedans une classe spécifique **BddtournoiDataContext**
Et enfin la chaine de connexion à la base de données **BddtournoiDataContextConnectionString**

→ Cliquez sur Next



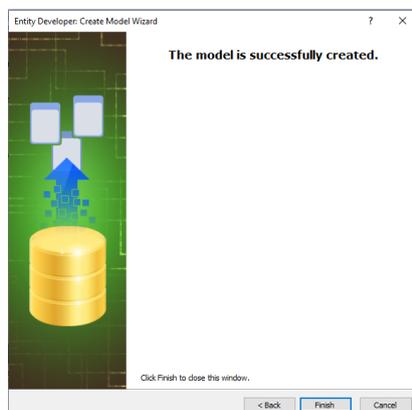
On choisit un diagramme de classes avec toutes les tables qui ont été sélectionnées précédemment

→ Cliquez sur Next

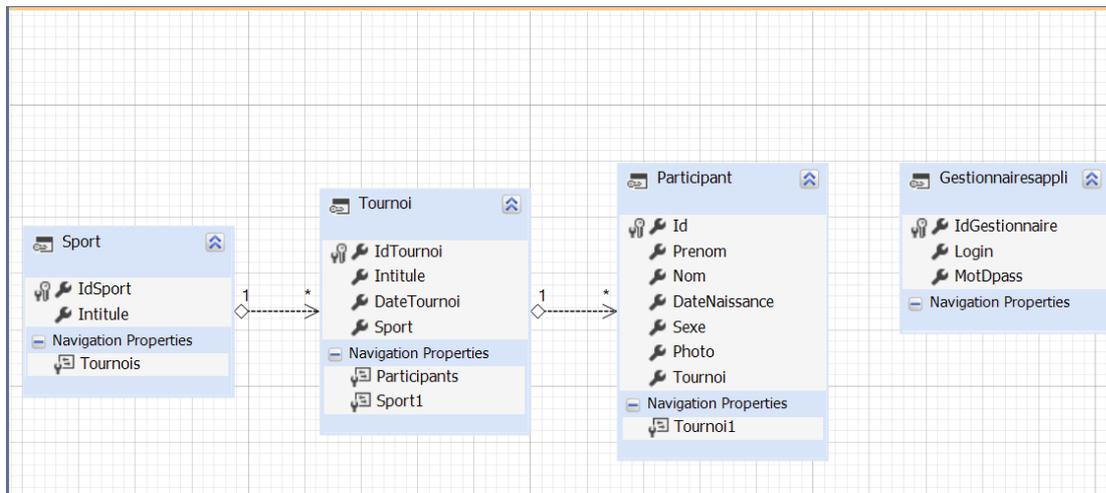


A cette étape, Linq donne des informations sur le code qu'il va générer

→ Cliquez sur Next

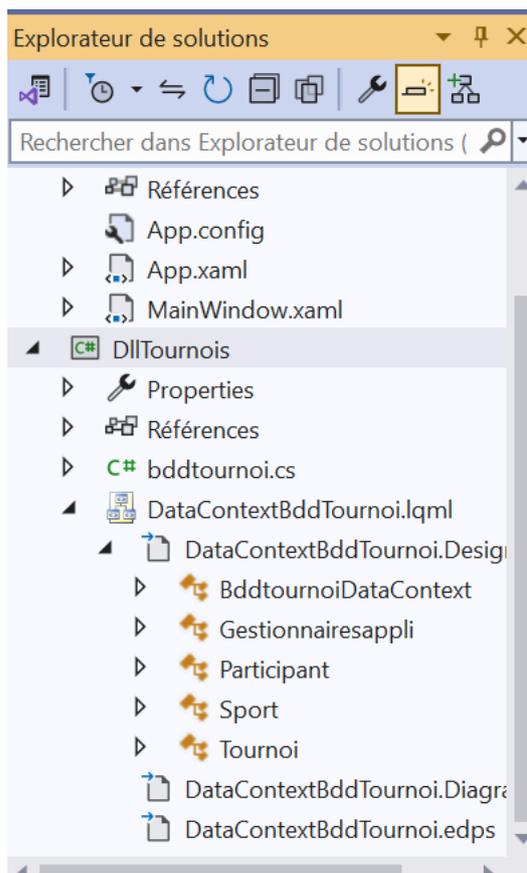


→ Cliquez sur finish, si tout s'est bien passé, les tables de la base de données s'affichent :



A cette étape, on peut voir que Linq a ajouté plusieurs références au projet et a généré toutes les classes représentant les tables de la base de données ainsi que le schéma de la base.

En déroulant "BddtournoiDataContext", on peut voir les méthodes générées :



*Maintenant il est possible de commencer à coder les méthodes de la classe **bddtournoi.cs**.*

Pour utiliser toutes ces méthodes généreusement créées par Linq, il est possible de se documenter dans la rubrique documentation (extraction de données) dont le lien est enregistré à la suite de l'installation de Linq. Un lien vers des exemples C# est également fourni.

→ A ce stade, tous les outils dont vous avez besoin sont prêts.

V. Développement :

1. Complétez votre projet **AppTounoi** avec toutes les vues dont vous avez besoin, par exemple :
 - La vue des paramètres de la base de données.
 - La vue de consultation et des différentes recherches
 - La vue d'identification pour utiliser l'application en tant que gestionnaire. *Cette vue apparaît en cliquant sur un MenuItem **Gestionnaire**, ce qui permettra d'activer les MenuItem nécessaires au gestionnaire (au lancement de l'application ces menus devront être désactivés).*
 - La vue pour gérer, un tournoi ou un sport.
 - La vue pour gérer un participant.
 - La vue pour gérer un gestionnaire d'application.

(Cet ensemble de vues (fenêtres) n'est qu'une proposition, vous pouvez mettre en place votre propre solution pour les fenêtres).

2. Complétez la classe **bddtournoi.cs** qui permet l'accès à la base de données en utilisant les méthodes générées par Linq. (Constructeur, méthodes de récupération des contenus des tables, méthodes d'ajout etc...)
3. Complétez les classes associées aux différentes fenêtres de votre projet afin de réaliser ce qui est demandé.

Pour la gestion des photos il est conseillé d'utiliser la classe **BitmapImag**. Une documentation et des exemples sont mis à disposition par Microsoft ainsi que des explications en cours.

L'utilisation du *binding* pour lier les vues aux données de la base de données est obligatoire.

Les *DataTemplate* sont également imposés pour la représentation des données dans des ListView ou DataGrid.

La stylisation de votre application sera également à évaluer par le professeur en charge du groupe de TD.

VI. Planning et évaluation :

- Séance 1 : (10pts)
 - ✓ Création de la base de données (1pt)
 - ✓ Création de la solution et des projets (1pt)
 - ✓ Ajout de l'outil Linq (1pt)
 - ✓ Création de la vue principale (vue de consultation) (3pts)
 - ✓ Création de la vue de configuration des paramètres de la base de données (1pt)
 - ✓ Connexion à la base de données (settings) (en affichant par exemple la liste des tournois : intitulé et date) (3pts)

→ Séance 2 : (10pts)

- ✓ Affichage de la liste des sports. (1pt)
- ✓ La sélection d'un sport permet d'afficher les tournois associés à ce sport (2pts)
- ✓ La sélection d'un tournoi permet d'afficher la liste des noms des participants. (2pts)
- ✓ Un menu "**Liste de tous les participants**" permet d'ouvrir une fenêtre pour afficher tous les participants avec l'ensemble des informations (nom, prénom, date de naissance, photo, sport et tournoi) dans un **DataGrid**. On peut alors trier par ordre_alphabétique des noms ou rechercher par tout ou partie du nom. (5pts)

→ Séance 3 : (10pts)

- ✓ Création de la vue d'authentification du gestionnaire (1pt)
- ✓ Test de l'authentification (menus du gestionnaire dégrisés) (1pt)
- ✓ Création de la vue de gestion d'un tournoi (1pt)
- ✓ Création de la vue de gestion d'un sport (1pt)
- ✓ Ajout d'un tournoi et d'un sport à la base de données (2pts)
- ✓ Création de la vue gestion de participants (1pt)
- ✓ Ajout d'un participant à la base de données (3pts)

→ Séance 4 : (7pts)

- ✓ Complétez les vues de gestion des tournois, sports ou participants afin de pouvoir modifier ou supprimer un tournoi, un sport ou un participant à l'aide d'un **ContextMenu** (2pts)
- ✓ Création de la fenêtre de gestion des gestionnaires de l'application (1pt)
- ✓ Ajout d'un gestionnaire (2pts)
- ✓ Modification ou suppression d'un gestionnaire à l'aide d'un **ContextMenu** (2pts)

→ Les allures des vues et stylisation : (3pts)

La validation se fera au fur et à mesure lors de la séance du TD. Les dernières 30mn ou 15 mn selon l'avancement sera réservé aux dernières validations de la séance.

Une aide du professeur est possible durant la séance pour ne pas rester bloqués ; un malus sera appliqué à cette aide selon l'importance de cette dernière.

NB : des exemples de vues sont proposés dans le fichier [ExemplesDeFenêtres.pdf](#)