

ENONCE DU PROJET

1. Le projet

Problématique :

Monsieur DUPONT se lance dans l'enseignement au sein de plusieurs collèges. Son objectif est de pouvoir gérer ses classes d'élèves ainsi que leurs notes et leurs absences afin gagner du temps dans la gestion administrative et de faire des compte-rendu réguliers au corps enseignant. Pour se faire, Monsieur DUPONT a imaginé un système complet avec les éléments suivants :

- Une base de données permettant de stocker toutes ses classes, ses élèves, leurs notes, leurs absences, etc.
- Une application installable sous forme d'un exécutable pour gérer le relevé de note c'est-à-dire l'ajout et la modification de notes. Cette application sera installée sur son poste et accèdera à la base de données.
- Une application web pour lui permettre de saisir les absences de ses élèves en temps réel mais aussi pour pouvoir consulter ou éditer n'importe quel bulletin de note rapidement (exemple : Conseil de classe).
- Un service web pour exposer ces précieuses informations à ses collègues désirant collaborer avec lui sur le même principe.

Solution envisagée :

Afin de répondre à tous les besoins de Monsieur DUPONT en utilisant exclusivement des technologies Microsoft, voici la solution à mettre en place :

- Une base de données SQL Server.
- Une bibliothèque de classe pour gérer la couche d'accès aux données (J1).
- Des tests unitaires pour ajouter des cas de test à la couche de données (J1) et assurer la pérennité de la couche de données.
- Une application WPF pour concevoir une application client lourd permettant de gérer les relevés de notes de ses élèves (J2).
- Une Web API pour gérer la synchronisation entre la base de données et un consommateur quelconque (une application mobile, un collègue via un navigateur, etc) (J3).
- Une application Web pour disposer d'un portail accessible de n'importe où et lui permettant de travailler depuis un de ses établissements.

2. Préambule

La base de données :

Afin de faciliter le développement et le déploiement de l'application, la base de données sera volontairement au format « **mdf** ». Nous ne travaillerons pas sur une base de données hébergée sur un serveur SQL. Cette méthode ne change en rien la méthode de développement et surtout évite l'installation et la configuration d'un serveur SQL. Ce fichier au format mdf sera à rendre avec la solution complète à la fin du TP.

Le schéma de la base de données :

Voici la base de données que Monsieur X a imaginée :

[SCHEMA A REALISER EN TD]

NB : Ce schéma pourra être amené à être complété au fur et à mesure du projet mais représente une base de travail indispensable à réaliser pour la suite du projet.

L'architecture de la solution :

A des fins de maintenance, Monsieur X souhaite que l'ensemble du projet soit stocké dans une seule et unique solution Visual Studio. La solution comprendra donc plusieurs « sous-projets ». Le découpage des sous-projets conseillé est le suivant :

- Un projet de type Bibliothèque pour la couche d'accès aux données.
- Un projet de type application WPF pour l'application permettant de gérer les relevés de notes.
- Un projet de type WEB API pour créer le service permettant la synchronisation de ses données.
- Un projet de type ASP.NET MVC pour créer l'application web permettant de consulter ses relevés et de saisir ses absences.
- Autant de projet de type Test Unitaire qu'il y aura de test à réaliser.

D'autres projets peuvent être ajoutés à cette solution pour compléter certains projets (exemple : un projet de type console pour tester la couche d'accès aux données dans un premier temps). La base de données au format présenté précédemment se placera dans le projet de la couche d'accès.