

ENONCE DU PROJET

1. Le projet

Préambule :

Le projet se déroulera sur toute la période par équipe de deux. Tout comme le cours, le projet sera découpé en 4 jalons (voir Readme du repository Cours). La notation se fera sur les critères suivants :

- Les fonctionnalités implémentées (et leurs bons fonctionnements).
- La qualité du code (commentaires, tests, normes de développement, nommage des variables, etc.).
- Le respect des consignes et de l'architecture du projet.
- La compilation (pas d'erreur, le moins de warning possible).
- L'ergonomie des interfaces.

Le projet sera à rendre à la fin de la période avec la base de données.

Problématique :

Une multinationale souhaite mettre à disposition de ses employés une structure intranet afin de diffuser des offres d'emploi en interne. Leurs objectifs : Permettre à leurs employés de se positionner sur des offres pour évoluer en interne voire changer de branche. A travers ce portail, chaque employé pourra se créer un profil et le renseigner avec ses formations, ses points forts, etc. Il aura aussi accès à un espace pour consulter les offres disponibles à un instant T et pour postuler sur une offre. Ce portail sera aussi le moyen pour le service RH de l'entreprise de créer les offres et de consulter les personnes qui sont intéressées par l'offre.

La solution retenue pour réaliser une telle plateforme est la suivante :

- Une base de données permettant de stocker tous les profils et toutes les offres.
- Une application installable sous forme d'un exécutable pour permettre au service RH de créer une offre, de visualiser l'ensemble des offres en cours et de visualiser l'ensemble des postulants par offre.
- Une application web pour permettre à chaque employé de se créer un profil, de l'alimenter et de postuler sur les offres actives du moment. Il devra aussi pouvoir consulter les offres sur lesquelles il a une postulation en cours.
- Un service web pour « communiquer » avec une autre application intranet déjà existante permettant de retourner l'identité des postulants par offre. Cela permettra aux responsables des différents services d'avoir une liste des employés intéressés par les offres et de planifier des entretiens avec eux.

Solution envisagée :

Afin de répondre à tous les besoins de Monsieur X 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).

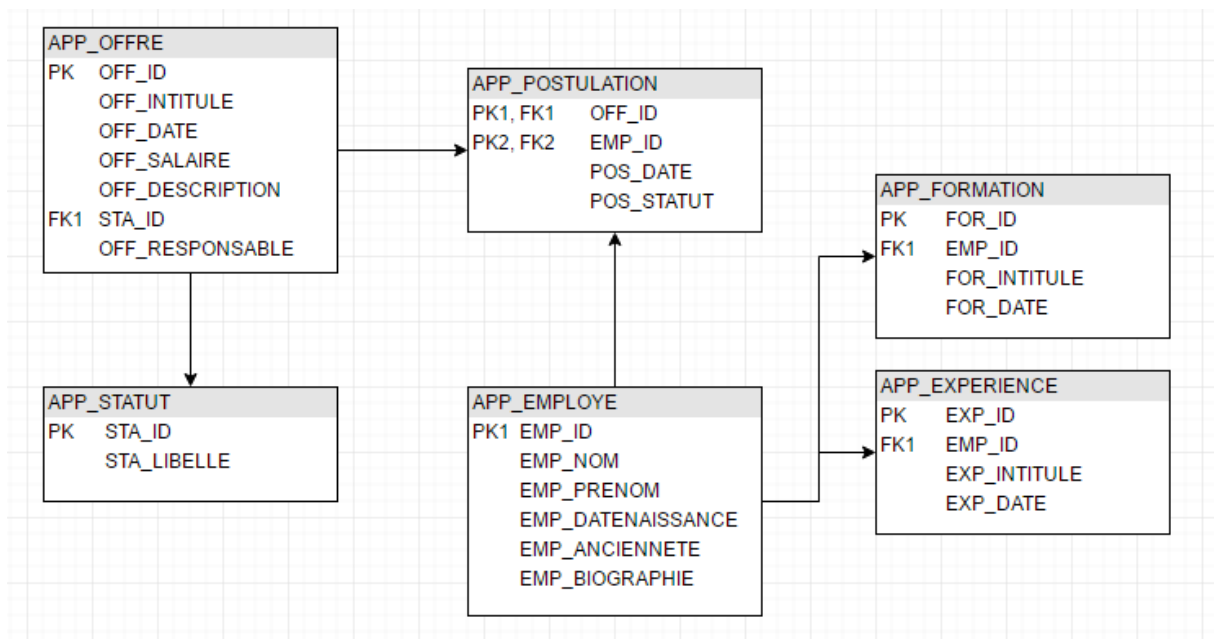
- Des tests unitaires pour ajouter des cas de test à la couche de données (J1) et assurer la pérennité de la couche de données.
- Une application WPF pour concevoir une application client lourd pour le service RH (J2).
- Un service WCF pour gérer la synchronisation entre la base de données et une application intranet « virtuelle » dédiée aux responsables des services (J3).
- Une application Web pour offrir aux employés la possibilité de consulter et de postuler sur une offre (J4).

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 :

La base de données imaginée pour l'occasion est disponible en annexe.



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, l'ensemble du projet doit être stockée 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 client lourd.
- Un projet de type application WCF pour créer le service permettant la synchronisation des données.

- Un projet de type ASP.NET MVC pour créer l'application web.
- 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.