



WineTeK

Choix fonctionnels et techniques

.NET Framework et langage C#

Le Framework .NET et le langage C# font partie des technologies qui ont été choisies pour la réalisation de cette application. En effet, de part la richesse du Framework .NET et l'orientation fortement objet de C#, ce langage tend à s'imposer dans le monde Windows comme la nouvelle référence.

SQL Server 2008

SQL Server 2008 est le SGBDR (Système de Gestion de Base de Données Relationnelles) de Microsoft. Ce dernier s'intègre très bien dans une solution Visual Studio C#, c'est pour cette raison que cette technologie a été choisie.

Entity Framework comme ORM

Entity Framework est l'ORM (Object-Relational Mapping) qui est utilisé pour effectuer l'ensemble des accès aux données de l'application. L'utilisation de cet ORM permet de manipuler les données de façon objet.

Comme l'ensemble de la documentation disponible sur la MSDN (Microsoft Developer Network) Library concerne la dernière version 5 de Entity Framework, nous avons décidé d'utiliser cette version. Nous l'avons installé par l'intermédiaire du package Nuget officiel de Microsoft. L'ensemble reste bien entendu compatible Visual Studio 2010, conformément à la demande Client.

Un modèle .edmx des données a donc été généré à partir de la base de données SQL Server 2008 et les classes membres associées également grâce au EF 5.x DbContext Generator.

Architecture N-tiers

Afin de garder une cohérence et une lisibilité dans le code de ce projet, l'architecture N-tiers a été adoptée. La solution est donc composée de 4 couches, soit 4 projets : UI, Business, Data et Entity. Seul le projet de la couche UI est exécutable. Les autres projets sont des bibliothèques de classes donc des fichiers .dll Windows. Pour la nomenclature des classes, nous avons choisi <NomDeTable>Controller.cs pour la couche Business et <NomDeTable>Data.cs pour la couche data.

Travail collaboratif avec Team Foundation Server 2010

Le travail collaboratif n'est jamais aisé. Il faut trouver des outils et des solutions pour pouvoir travailler à plusieurs, de préférence en même temps sur des fichiers communs.

De nombreuses solutions de synchronisation de fichiers dans le cloud existent désormais tel que Skydrive, Dropbox ou encore Google Drive. Cependant, ces solutions n'apportent pas de réelle flexibilité lorsqu'il s'agit de versionner des projets entiers de développement. De plus avec ces différents outils les accès concurrents aux fichiers créent des copies en conflit.

Avec l'environnement de développement intégré de Microsoft : Visual Studio, une solution complémentaire est disponible pour répondre à cette problématique : Team Foundation Server. Cet outil, entièrement intégré à Visual Studio permet à plusieurs collaborateurs de travailler sur un même projet en gardant une trace de l'ensemble des modifications effectuées. L'outil donne la possibilité de revenir à une révision antérieure, ce qui permet d'annuler des modifications si nécessaire.

Dans le cas où plusieurs développeurs modifient le même fichier en même temps, plusieurs cas peuvent se poser :

- Le fichier est un fichier texte (code source) et les modifications peuvent automatiquement être fusionnées, il suffit de cliquer sur un bouton.
- Le fichier est un fichier texte (code source) mais les modifications ne peuvent pas être fusionnées automatiquement, le développeur doit utiliser l'outil de fusion pour choisir quelles modifications conserver manuellement.
- Le fichier est binaire et l'utilisateur doit choisir le fichier à conserver : version du fichier local ou version du fichier du serveur.

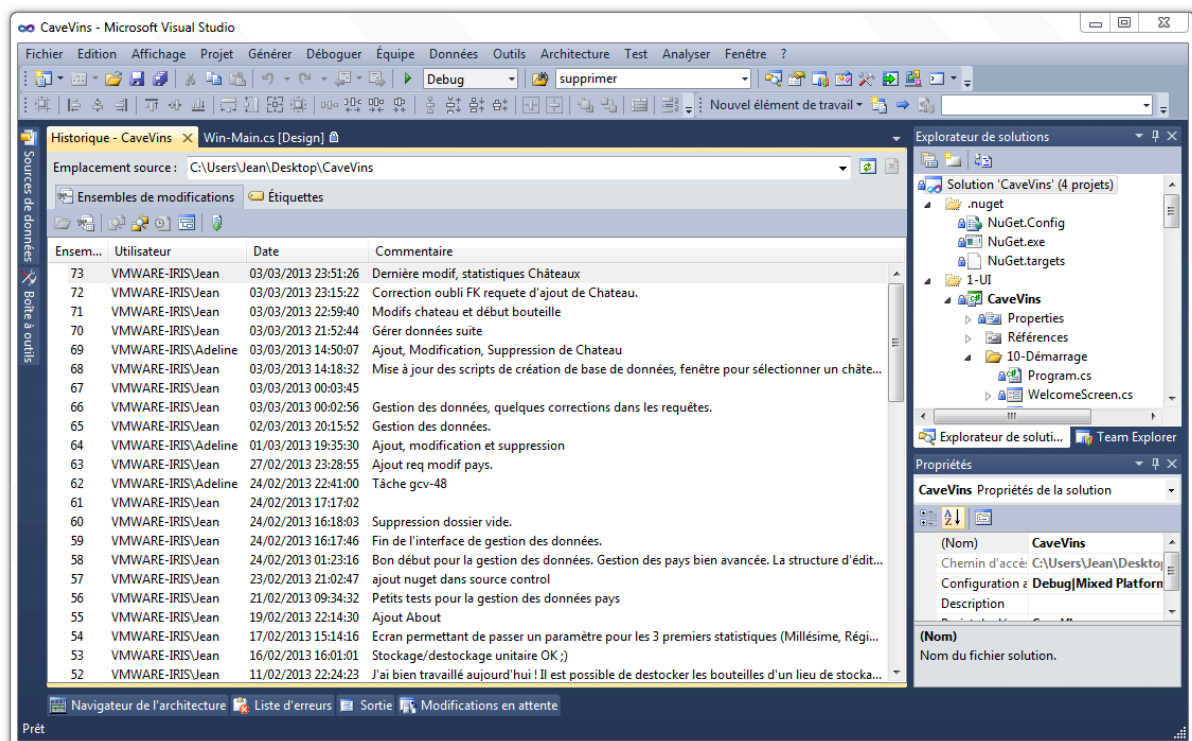


Figure 1 - Écran présentant les dernières révisions du projet WineTeK (Team Foundation Client)

Team Foundation Server est un excellent outil. Il nous aura permis de gagner du temps de façon à nous concentrer sur le développement de l'application plutôt que sur de la transmission d'informations.

Extension du modèle de données

Pour répondre au cahier des charges, nous avons été amené à étendre le modèle de données actuel pour lui adjoindre de nouvelles tables :

- T_INVENTAIRE_INV
- T_LIEU_LIE
- T_EMLACEMENT_EMP
- T_STOCKAGE_STO

Le schéma ci-dessous présente la nouvelle structure de données et met en évidence les nouvelles tables ajoutées.

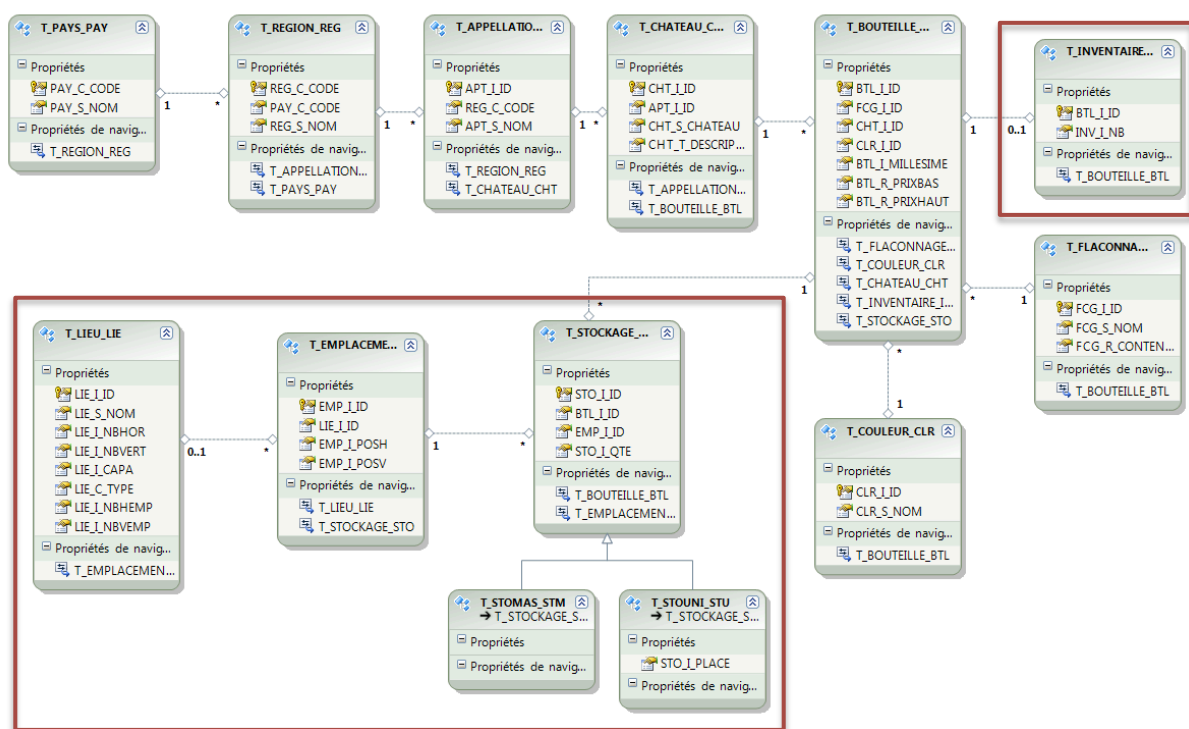


Figure 2 - Structure de données (les tables ajoutées sont encadrées en rouge)

Utilisation de l'héritage pour le stockage des bouteilles

Le stockage des bouteilles varie en fonction du fait que la référence est stockée en masse ou à l'unité. Si la référence est stockée en masse, elle ne possède pas de position dans un emplacement et la quantité de bouteille stockée est forcément supérieure à un. En revanche, si la référence est stockée dans un stockage unitaire, la quantité doit être forcément égale à un et la bouteille doit avoir une position dans l'emplacement.

Nous avons donc créé deux classes qui héritent de T_STOCKAGE_STO :

- T_STOMAS_STM : pour le stockage de masse
- T_STOUNI_STU : pour le stockage unitaire

On veillera bien à définir la classe T_STOCKAGE_STO en tant que classe abstraite. En effet, on ne doit pas pouvoir l'utiliser directement. Seules les classes héritées seront utilisées.

Interface utilisateur

MDI – Multi Document Interface

Nous avons choisi d'utiliser le Multi Document Interface. Cette technologie nous permet d'afficher l'ensemble des formulaires de l'application depuis une seule fenêtre de façon à garder une certaine cohérence. De fait, la barre de menu principale de l'application reste affichée quel que soit le formulaire présent à l'écran ce qui permet à l'utilisateur de naviguer plus simplement entre les différents écrans de l'application.

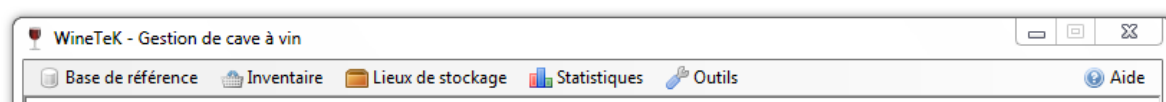


Figure 3 - Barre de menu principale de l'application

Affichage des lieux de stockage

Comme nous souhaitons garder l'expérience utilisateur et la simplicité d'utilisation au centre de cette application, il nous a semblé naturel de disposer d'un affichage des lieux de stockage. Cette présentation « visuelle » des lieux de stockage est utilisée pour effectuer différentes tâches dans l'application parmi lesquelles :

- Afficher un lieu de stockage de masse
- Afficher un lieu de stockage unitaire
- Choisir dans quel emplacement d'un lieu stocker des bouteilles en masse
- Choisir à partir de quel emplacement d'un lieu supprimer des bouteilles en masse
- Choisir à quelle position d'un emplacement d'un lieu de stockage stocker une bouteille
- Choisir une bouteille à supprimer d'un emplacement d'un lieu de stockage (unitaire)

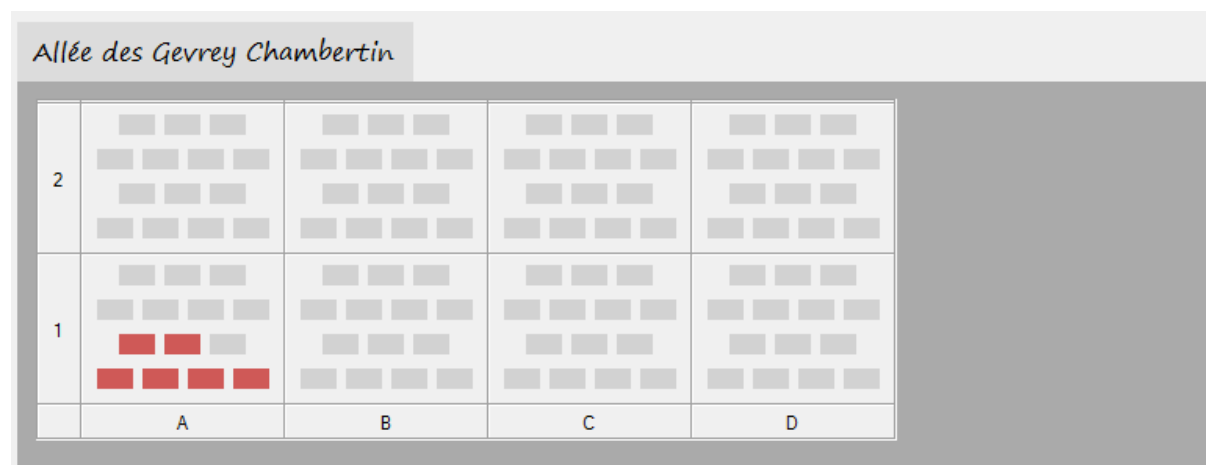


Figure 4 - Écran de l'application affichant un lieu de stockage

Les lieux de stockage sont représentés par des TableLayout Panels imbriqués les uns dans les autres et sont générés à la volée lors de l'affichage. Les bouteilles sont représentées par des panels qui sont colorés lorsqu'une bouteille est présente. Lors du survol sur une bouteille, une info-bulle s'affiche, détaillant le château et millésime de la bouteille.

Déploiement simplifié avec ClickOnce

Afin de simplifier au maximum la procédure d'installation et d'améliorer l'expérience utilisateur, nous avons choisi d'utiliser la solution de déploiement ClickOnce de Microsoft. Cette technologie possède l'avantage indéniable de vérifier et de télécharger le cas échéant les composants requis pour l'exécution de l'application si ces derniers ne sont pas présents sur le poste de l'utilisateur.