RAPPORT DE STAGE DE MASTER M2 INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE
UNIVERSITÉ DE LA RÉUNION
ANNÉE UNIVERSITAIRE : 2013-2014

Développement d'une application ASP.NET avec Base de données SQL Server

Auteur :
Emmanuel SAMBASSOUREDY
31001648

Encadrants :
M. Alain DORSEUIL
M. Hugues ROUGEMONT

JUIN 2014

Remerciements

Je tiens à remercier dans un premier temps mes encadrants, M. Alain DORSEUIL et M. Hugues ROUGEMONT, de m'avoir accueilli dans la société et pour m'avoir permis de réaliser ce stage, ainsi que pour leurs conseils sur les objectifs à atteindre.
Je tiens à exprimer ma gratitude au stagiaire de l'école SUPINFO Anthony PICARD, au développeur Esteban SARO qui ont su maintenir une ambiance propice au travail et à l'échange de connaissances.

Je remercie également les autres membres de run [in] box, M. Dany BIGEY , M. Yoann CAVALLIN, M. Vincent HERLIN et Mme Aurélie ZAGARIA, pour leur accueil.

J'exprime ma profonde gratitude à toute l'équipe enseignante de l'Université de la Réunion qui m'ont permis d'arriver à ce niveau d'étude.

Résumé

Dans le cadre d'une refonte d'un outil informatique de gestion de la sécurité et de la qualité, une société de stockage et distribution de produits hydrocarbures dangereux a missionné la société run [in] box pour le développement d'une application de type Web autour des technologies Microsoft ASP.NET avec une gestion des données sous le moteur SGBD SQL Server.
Le présent rapport a pour but de présenter le contexte métier du développement en passant par un rapide portrait de la société run [in] box, un récapitulatif des technologies existantes permettant la réalisation avancée d'application Web hautement interactive, pour finir par les aspects techniques mise en œuvre pour la conception des modules de traitement et de l'interface utilisateur.

Mots clés : Application web, Framework .NET, ASP.NET WebForms, ViewState, SQL Server, M-Files, jQueryUI

Abstract

In the context of an overhaul of a software tool for managing the safety and quality, a storage and distribution of hazardous products company has commissioned the company run [in] box for the development of a web-based application around the Microsoft ASP.NET technology using SQL Server DBMS engine as data management .
This report aims to present the context of software development through a quick picture of the company run [in] box, then by a summary of existing technologies for highly interactive Web application implementation and finally the technical implementation for the design of the processing modules and the user interface.

Keywords : Web-based application, .NET Framework, ASP.NET WebForms, ViewState, SQL Server, M-Files, jQueryUI

Table des matières

Table des figures

1 Introduction

Ce rapport de stage est écrit par Emmanuel SAMBASSOUREDY dans le cadre du stage de fin d'étude du master informatique de l'Université de la Réunion. Ce stage a été encadré par :

— M. Alain DORSEUIL, Directeur de la société run [in] box.
— M. Hugues ROUGEMONT, Directeur technique de la société run [in] box.

Les technologies de l'information et de la communication font partie intégrante de notre quotidien. En effet, le monde est de plus en plus connecté, que ce soit par les ordinateurs de bureau, ordinateurs portables ou encore des smartphones ou tablettes. L'avènement de ces dispositifs qui ont la particularité d'être simples d'utilisation et pratiques à transporter, ont favorisé l'évolution du développement des applications informatiques vers les technologies orientées Web. De plus, celles-ci trouvent également une large application dans le monde de l'entreprise où il faut pouvoir développer, déployer et maintenir rapidement des outils informatiques sur un nombre important de postes utilisateurs sur différentes plateformes. Ainsi, beaucoup d'entreprises se tournent vers ce genre d'application et beaucoup d'éditeurs mettent au point différents frameworks aidant à la conception rapide et efficace d'outils métiers.

Ce stage a été l'objet de la refonte d'une application vieillissante qui n'est plus maintenue, vers des technologies modernes et orientées Web. Cette demande a été émise par une entreprise qui assure la gestion et le stockage de matières dangereuses à l'île de la Réunion. L'objectif de cette application est d'améliorer la réactivité des salariés de l'entreprise face aux dangers qui peuvent intervenir dans leurs quotidien. En effet, celui-ci est destiné à apporter des solutions en amont des problèmes pour préserver les risques d'accident ainsi que leurs conséquences. Ma mission principale durant ces six mois a été de réaliser cette refonte en application web.

Dans un premier temps, je vais décrire l'établissement qui m'a accueilli, ses activités ainsi que ces objectifs, puis dans un second temps, je présenterai en détail la mission qui m'a été confiée par l'entreprise, où je montrerai les différents objectifs fixés, ainsi que la méthodologie appliquée pour le développement du projet. Je continuerai ensuite par les différents approfondissements effectués, notamment un aperçu de la méthodologie appliquée pour la réalisation de l'application, mais aussi les difficultés rencontrées. Pour conclure, je dresserai un bilan de ce qui a été réalisé et présenterai les perspectives d'avenir du projet.

2 L'entreprise d'accueil

2.1 run [in] box

run [in] box, Société de Services en Ingénierie Informatique créée en 2003, est spécialisée dans la gestion de flux documentaires sécurisés, l'hébergement, l'infogérance et l'intégration de solutions décisionnelles et collaboratives. Initialement créée pour répondre à une stratégie de mutualisation informatique dans le secteur de la Santé, la SS2I run [in] box adresse aujourd'hui l'ensemble des secteurs d'activités de l'île.

2.1.1 Ses activités

La société propose des solutions d'hébergement sur mesure pour divers secteurs d'activités tels que :

- Les professionnels de santé
- La grande distribution et commerces divers
- Les banques, les assurances et le juridique
- Les collectivités et les grands comptes

2.1.2 Ses objectifs

Basée à la Réunion, l'entreprise travaille avec différents partenaires logiciels européens. De plus, elle s'est engagée dans un processus continu d'amélioration de la qualité avec comme objectif final la satisfaction du client.
Elle s'engage donc à assurer :

La qualité et la sécurité
L'entreprise est dans une démarche de processus Qualité selon la norme ISO 9001 [1].

La confidentialité
run [in] box est à ce jour la seule SS2I réunionnaise à avoir obtenu un agrément du Ministère des Affaires Sociales et de la Santé pour l'hébergement de données de santé à caractère personnel. Cette démarche d'agrément a notamment permis de proposer à d'autres secteurs d'activité que la Santé des conditions de confidentialité bien au-delà des standards du marché de l'hébergement classique.

1. ISO 9001 : La norme ISO 9001 définit une série d'exigences concernant la mise en place d'un système de management de la qualité dans un organisme.

Une haute-disponibilité des données

Le datacenter de run [in] box est constitué de deux sites distants en mode actif-actif et reliés par fibre optique. Différents niveaux de SLA peuvent être proposés en fonction des nécessités des applications et de la typologie de clients.

run [in] box propose sept niveaux d'infogérance, le niveau [in] 7 correspondant par exemple à l'exploitation d'applications en mode SaaS [2]. Le niveau [in] 0 des services d'infogérance de run [in] box consiste à la mise à disposition des clients d'une infrastructure offrant :

- De l'énergie, via des réseaux électriques séparés, ondulés et secourus par deux groupes électrogènes
- Des unités de climatisation redondantes
- Des dispositifs de sécurité incendie
- Un accès sécurisé : contrôle d'accès avec badge sans contact et système d'identification biométrique, sécurité anti-intrusion avec supervision par vidéo surveillance
- Des accès réseau
- Une l'infrastructure d'hébergement informatique

La figure n° 1 illustre l'infrastructure du datacenter de l'entreprise (une version détaillée de ce schéma est disponible en annexe à la page n° 46).

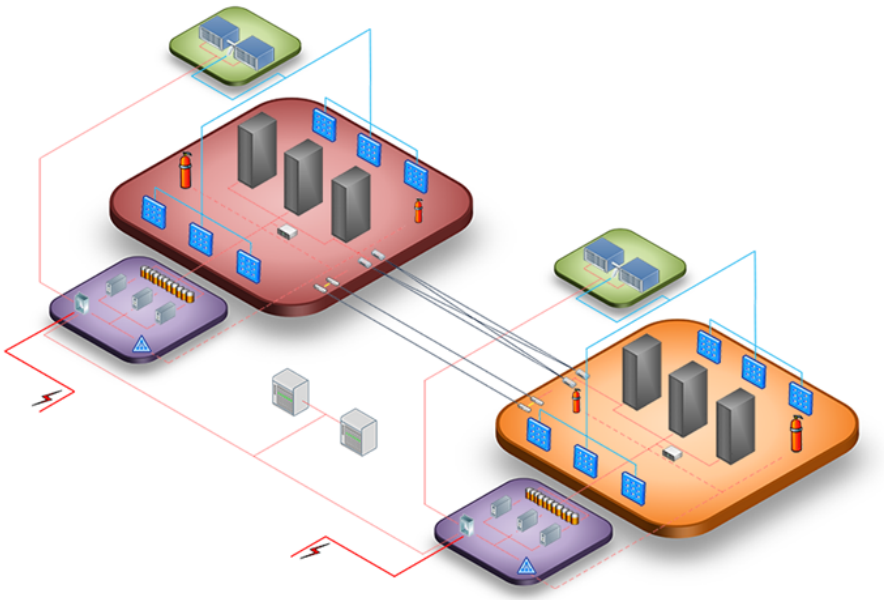


FIGURE 1 – Infrastructure du datacenter de l'entreprise

2. Software as a Service (SaaS) : Il s'agit d'un modèle d'exploitation des logiciels, où ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur.

Au-delà de l'hébergement & infogérance, run [in] box propose différentes solutions axées sur :

— L'informatique décisionnelle [3]
— L'automatisation des flux documentaires
— La gestion et le suivi de projets en mode collaboratif

run [in] box propose également une solution de coffre-fort électronique à valeur probante qui peut être mise en œuvre dans le cadre des projets décisionnels et collaboratifs.

2.2 L'équipe

L'équipe compte actuellement sept personnes organisée selon trois pôle essentiels : commercial, développement/infogérance et datacenter (illustrés par la figure n° 2). L'entreprise répond aux problématiques des client selon trois axes :

— La réponse aux problématiques « métier »via des solutions décisionnelles,
— La réponse aux problématiques de gestion de projets et de travail collaboratif des équipes,
— La réponse aux problématiques de valorisation du patrimoine documentaire de l'entreprise (GED, valeur probante, etc.).

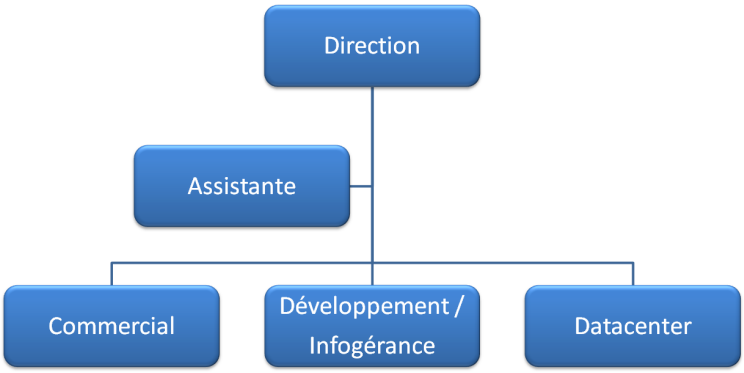


FIGURE 2 – Organigramme de run [in] box

Durant ce stage j'ai intégré le pôle développement/infogérance.

3. Informatique décisionnelle : Il s'agit de l'informatique à l'usage des décideurs et des dirigeants des entreprises. Elle désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données, matérielles ou immatérielles, d'une entreprise en vue d'offrir une aide à la décision et de permettre à un décideur d'avoir une vue d'ensemble de l'activité traitée.

3 La mission

Ce stage s'inscrit dans un projet de refonte d'un logiciel de type client/serveur vers une application web accessible en intranet. Cette refonte a été formulée par une entreprise de gestion et de stockage de matières dangereuses. Elle permettra aux salariés de cette entreprise d'améliorer leur rendement tout en garantissant un niveau de sécurité permanent. En effet, la nature des produits qu'elle stocke et qu'elle manipule l'a conduite à faire de la sécurité une priorité. Elle déploie donc beaucoup d'efforts pour maintenir son niveau de compétence et sensibiliser ses effectifs pour que tout se déroule dans les meilleures conditions. Ainsi, il a été mis en place un processus de remontée et d'analyse des évènements indésirables pour se prémunir des conséquences de situations dangereuses qui pourraient provoquer un accident. Ce processus a pour objectif d'apporter des solutions en amont des problèmes pour éviter ou mieux maîtriser ces situations. Un processus de suivi d'indicateurs de sécurité tels que le nombre de jours sans accident, ou le nombre de remontée d'évènement par exemple a également été mis en place.

Le client dispose d'un logiciel de gestion des indicateurs de performance développé sur mesure qui permet :

— La gestion et la valorisation d'une base d'indicateurs relatifs à la sécurité.
— La gestion des rapports d'accident et le suivi du processus de recommandations qui en découlent.
— La gestion des rapports d'audit et leur questionnaire.

Cette solution fonctionne en mode client/serveur et repose sur les technologies suivantes :

— SGBD-R Oracle 8.1 pour la base de données.
— Visual Basic 6.0 pour les interfaces réalisées.
— Business Objects 5.1.2 pour la partie informatique décisionnelle.

Un complément Web à ce logiciel a été ensuite réalisé et intégré au réseau intranet de l'entreprise. Celui-ci permet le suivi des performances ainsi que la saisie des recommandations propres à l'entreprise cliente.
Bien que la solution n'aie pas évolué depuis et n'est pas couverte par un contrat de maintenance, elle reste tout de même utilisée par trois utilisateurs principaux et par environ 40 opérateurs en intranet.

Un cahier des spécifications techniques et fonctionnelles a été fourni à la société run [in] box expliquant en détail l'expression des besoins de l'entreprise concernant cette nouvelle application.
J'ai été chargé de mener à bien l'aspect développement sur le projet au sein de l'entreprise run [in] box.

3.1 Cahier des spécifications

Pour réaliser ce projet, un cahier des spécifications techniques et fonctionnelles a été fourni par le client. Ce document constitue une synthèse de l'audit technique et décrit toutes les prestations attendues dans le cadre du projet de refonte de l'application. Il présente les besoins fonctionnels ainsi que l'ensemble des règles techniques à respecter lors de la réalisation.
Celui-ci contient :

- Les contraintes fixées par le client
- La nouvelle architecture de données
- Les maquettes des futurs écrans du logiciel transactionnel accompagnées de leurs règles de gestion
- Une représentation schématique des calculs des indicateurs de performance
- Les éléments initiaux pour la mise en place de l'outil de Business Intelligence
- Un plan de migration
- Les prestations attendues du soumissionnaire

3.2 Contraintes

Le client a listé dans le cahier des spécifications ses attentes pour la nouvelle solution :

- Un seul logiciel transactionnel en mode Web : réalisation de la nouvelle solution avec les outils de dernière génération :
  - Interfaces à réaliser en mode « Web »regroupant tous les écrans de la solution existante (« client / serveur »et « Intranet ») dans un seul « noyau ».
  - Usage de l'outil de SGBD-R Microsoft SQL Server
  - Usage d'un outil de Business Intelligence de dernière génération
  - Compatibilité avec les systèmes d'exploitation Windows Server 2008 R2, Windows Server 2012 et Windows Server 2012 R2
  - Développement des interfaces graphiques en mode « Web »à l'aide un langage de programmation différent du langage JAVA
  - Compatibilité avec les navigateurs standards du marché (Internet Explorer 8 à 10, Google Chrome, Firefox,. . .)

— Des écrans répondants aux besoins actuels, dynamiques et évolutifs : les écrans seront toujours axés sur les concepts « métiers », auxquels s'ajouteront de nouvelles fonctionnalités et « ouverts »aux évolutions futures. Les nouveaux écrans seront plus conviviaux, intuitifs et faciles d'utilisation.

— Une maintenance corrective, adaptative et évolutive de la solution globale : s'appuyer sur un prestataire en cas de problème sur l'application. Par exemple, lorsque l'application ne fonctionne plus ou lorsqu'il s'agit de faire évoluer le logiciel :
  — d'un point de vue fonctionnel selon les corrections ou les évolutions métier à apporter
  — d'un point de vue technique afin de suivre les évolutions technologiques (tels que les tests de compatibilité du logiciel sous une nouvelle version de système d'exploitation, de base de données et adaptations éventuelles pour que l'application fonctionne sous des versions plus récentes de logiciels)

3.3 Objectifs

Pour mener à bien ce projet, plusieurs objectifs ont été identifiés :
— Définir une plateforme cible pour l'exécution de l'application,
— Développer l'ensemble des fonctionnalités de l'application,
— Créer les procédures stockées pour le traitement des indicateurs
— Mettre en œuvre d'une structure de base de données suivant un modèle
La figure n° 3 illustre le plan de l'application ainsi que les différentes fonctionnalités à développer.

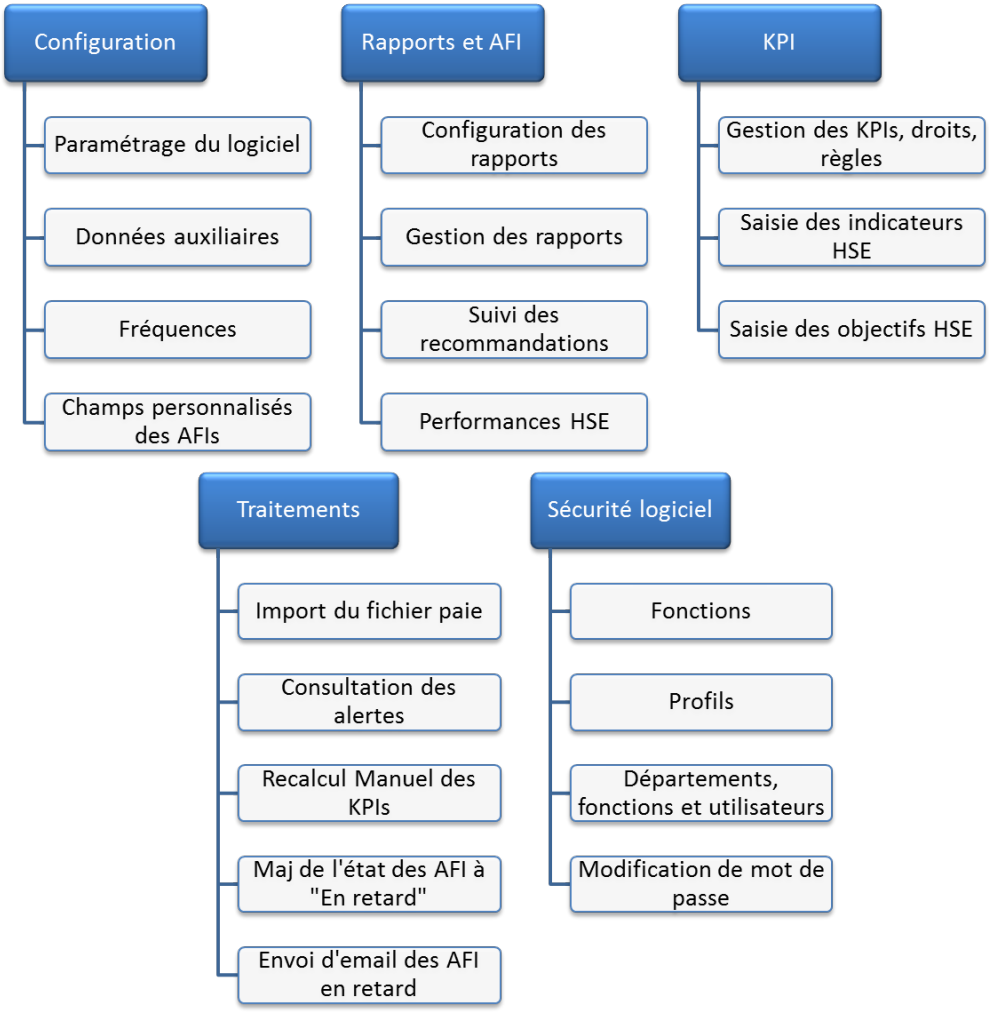


FIGURE 3 – Plan de l'application et de ses fonctionnalités

3.4 Gestion de projet

Durant ce stage j'ai été amené à travailler en coopération avec un stagiaire issu de l'école informatique SUPINFO sous la coordination du directeur de run [in] box, M. DORSEUIL Alain.

3.4.1 Processus de développement du projet

Le développement du projet se repose sur une approche agile. En effet cette approche permet au client d'avoir régulièrement un rendu des fonctionnalités de l'application tout au long de la durée du projet. La figure n° 4 offre un aperçu de cette approche. Chaque étape du processus est définie en concertation avec le client et le planning des livrables correspondants est défini conjointement.

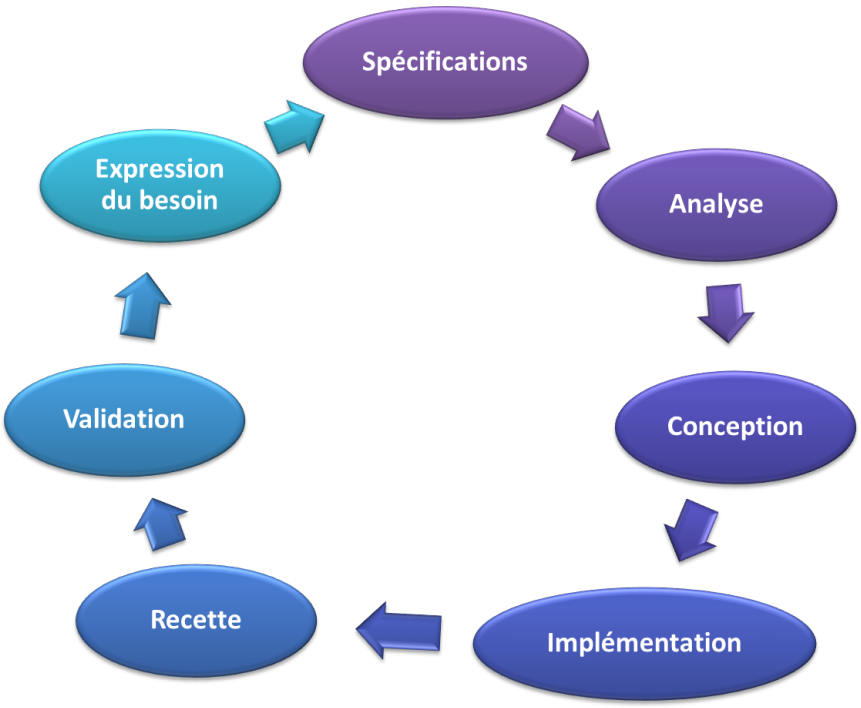


FIGURE 4 – Processus mis en œuvre pour le développement de l'application Web

Les caractéristiques des principales de ces phases sont :

— Spécifications fonctionnelles :
Recueil des besoins en fonction des différents acteurs sur site jusqu'à validation client. Mise en œuvre d'outils de travail collaboratif

— Modélisation (Analyse / Conception) :
Présentation sous forme de diagrammes des interfaces de saisies, des unités de traitement et des états.

— Tests et validation :
Validation sur la base d'un document de recette issu des spécifications

L'intérêt de cette démarche est de procéder à des livraisons successives nous permettant de rester en contact permanent avec les principaux utilisateurs du logiciel en cours de déploiement. Cette démarche permet d'aboutir à une meilleure satisfaction client qu'une approche classique de type Audit / Cahier des charges.

3.4.2 Gestion des éléments de recettes

Dès la mise à disposition d'une itération du livrable, il est essentiel de procéder à des vérifications de manière à contrôler la conformité du résultat avec le cahier des spécifications. Pour chaque itération, le client effectue des tests de l'application et communique la liste des anomalies rencontrées, ou encore, des changements non présents dans le document des spécifications lors du test, via le logiciel M-Files. A la prochaine itération, le client prendra le soin de vérifier que tous les éléments de recette traités soient conforme à ses attentes. La figure n° 5 présente une vue d'ensemble de l'écran de gestion de ces éléments où client peut créer ou définir l'état d'un élément de recette.

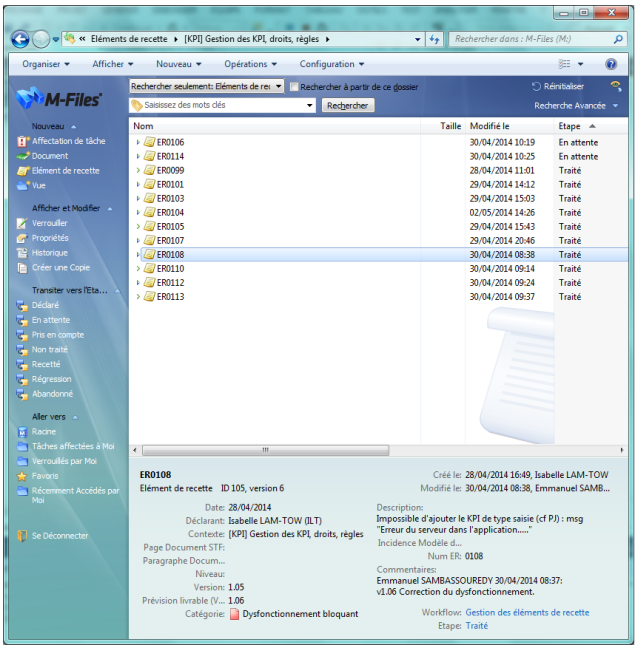


FIGURE 5 – Aperçu de l'écran des éléments de recettes du logiciel M-Files

Ces éléments sont classés en quatre catégories :

Ecart STF

Un élément de recette est placé dans cette catégorie, s'il ne respecte pas une règle de gestion mentionnée dans le cahier des spécifications techniques et fonctionnelles. Cela est généralement dû à une mauvaise compréhension de ces règles.

Correctif STF

Les éléments de recette de ce type sont, comme leur nom indique, des corrections de certaines règles de gestion formulées dans le document, apportées par le client.

Evolution STF

Comme nous l'avons dit précédemment, le contenu du cahier des spécifications techniques et fonctionnelles peut subir des modifications ou des ajouts de fonctionnalités supplémentaires sur la demande du client. Cette catégorie concerne tous les éléments de recettes liés à ces ajouts.

Dysfonctionnement bloquant

Il peut arriver que certaines fonctionnalités encore en cours de validation, génèrent des messages d'erreurs. Par sécurité les exceptions non gérées bloquent la page en cours, cela permet au reste du code de ne pas continuer à s'exécuter. Lorsque cela arrive, un élément de recette est créé dans cette catégorie.

3.4.3 Workflow

Afin d'optimiser la communication de l'avancement des éléments de recette avec le client, le logiciel M-Files fournit un système de Workflow [4] paramétrable. Le statut des éléments de recette suit le cheminement montré sur la figure n° 6 :

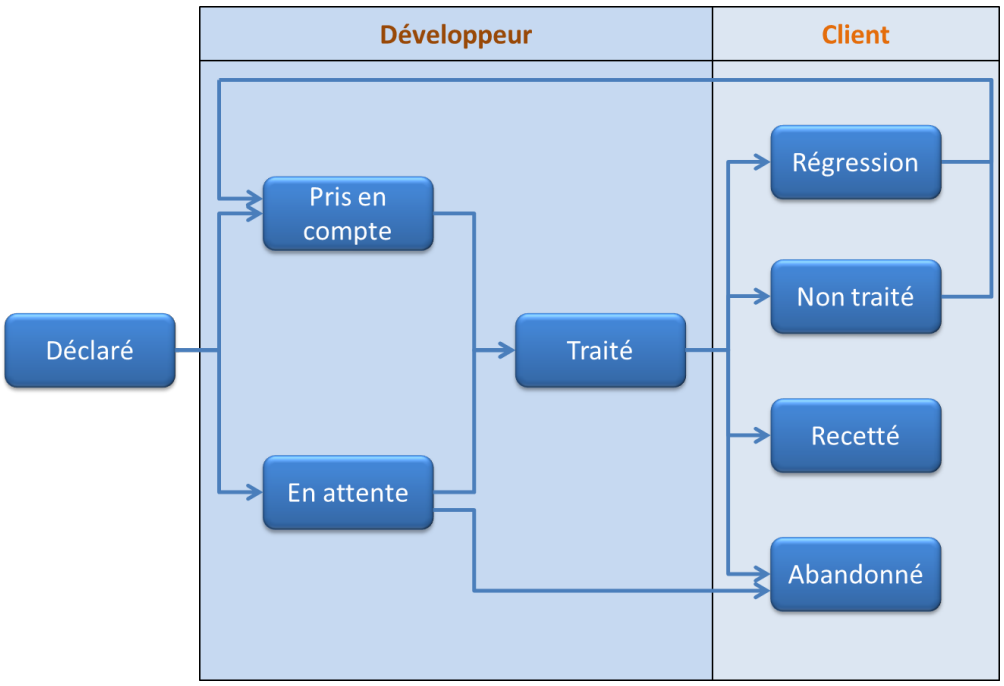


FIGURE 6 – Aperçu du cheminement du changement d'état d'un élément de recette

Chaque élément de recette posté dans l'application M-Files possède l'état *Déclaré*. Une fois que le développeur a pris soin de lire la description de l'élément, il peut soit le passer dans l'état *Pris en compte* pour indiquer qu'il travaille sur le sujet, ou soit le passer en *en attente* avec un commentaire dans le cas où il n'a pas toutes les informations relatives à élément (tel que le manque de précision lors de la déclaration d'un *Écart STF* par exemple). Une fois que les informations supplémentaires sont apportées par le client, l'élément de recette passe alors à l'état *Pris en compte* et le développeur vérifie s'il est possible de traiter l'élément. Si l'élément est traitable, alors le développeur effectue le traitement demandé et passera l'élément à l'état de *Traité* . En revanche, si la description donnée par le client n'est pas assez précise, l'élément concerné sera mis en attente avec un commentaire indiquant au client de fournir plus de précision sur la nature du problème. Une fois que les informations supplémentaires sont apportées, le développeur peut alors effectuer le traitement de l'élément.

Une fois que l'élément de recette est traité, il ne reste plus qu'au client de le valider ou non, en cas de validation, l'élément reçoit l'état *Recetté* sinon, il passe à l'état *Non traité* si la demande du client n'a pas été satisfaite. L'élément repassera alors en *Pris en compte* et sera retraité par le développeur.

4. Workflow : Il s'agit de la représentation d'une suite de tâches ou d'opérations effectuées par une ou plusieurs personnes.

Il est possible que la mise en place d'une fonctionnalité perturbe le fonctionnement d'une autre ou que la correction d'une erreur entraîne la création d'un autre dysfonctionnement. Si le client observe un de ces deux cas pour un élément *Recetté* , il doit changer l'état de celui-ci en *Régression* pour indiquer aux développeurs qu'il s'agit d'un dysfonctionnement présent sur un élément de recette déjà validé.

4 Approfondissements

4.1 Contexte technologique

Dans cette partie, je vais dresser un inventaire des différentes technologies existantes pour la mise en œuvre d'applications web.

4.1.1 Définition d'une application web

Une application web est une application qui s'exécute par le biais d'un navigateur Web et donc développée par un langage de programmation compatible avec les navigateurs côté client (telle que la combinaison du JavaScript, HTML et du CSS) et exécutable au niveau d'un serveur HTTP côté serveur.

Les applications web se sont popularisées grâce à la forte présence des navigateurs web dans le monde et leur facilité d'utilisation. La possibilité de maintenir à jour les applications web, sans perturber les milliers d'utilisateurs, est la raison principale de cette popularité tout comme la compatibilité intrinsèque entre les différents systèmes d'exploitations.

Dans la réalisation d'une application web, la gestion d'un contexte utilisateur est essentielle si on souhaite offrir une expérience personnalisée lors de leur navigation sur les pages d'un site. Cela permet à l'application de déterminer par exemple avec quel utilisateur elle communique, et ainsi, d'afficher à l'écran des informations auxquelles d'autre utilisateur n'ont pas accès.

4.1.2 Langages et frameworks

Dans cette partie je vais présenter différents langages de programmation qui pourraient être utilisés dans le cadre du développement d'application web, ainsi que quelques frameworks qui permettraient de répondre aux besoins.

Un framework (littéralement « Cadre de travail »en français) est un ensemble de bibliothèques [5] conçus dans le but de simplifier le travail des développeur en leur fournissant les composant nécessaires pour créer les fondations, l'architecture et les grandes lignes d'un logiciel. Il s'agit d'une boite à outils réalisée par des développeurs pour des

5. Bibliothèque logicielle : une bibliothèque logicielle est une collection de fonctions, qui peut être déjà compilée et prête à être utilisée par des programmes.

développeurs. Cependant, cette boite à outils n'est pas conçu pour qu'on l'utilise tel quel, il est nécessaire d'utiliser un langage de programmation pour pouvoir exploiter tous ses éléments.

Le principal objectif d'un framework est d'améliorer la productivité des développeurs : il leur offre la possibilité d'utiliser tel ou tel composants pour plus d'aisance dans le développement du logiciel. Par exemple, la récupération de données depuis une base de données et son affichage dans un tableau en HTML est un des composants que fourni le framework .NET. Cela permet de ne pas constamment réinventer la roue et de gagner du temps pour le reste du développement.

Un autre objectif du framework est de rendre le code source structuré, lisible et réutilisable par d'autres développeurs. De plus, un framework doit être souple et modulable pour pouvoir être utilisé dans différents types de projets.

Comme nous l'avons dit précédemment, le principal avantage de l'utilisation d'un framework est le gain en productivité, le code est organisé et lisible. De plus les composants du framework permettent au développeur de ne pas se répéter dans son code pour effectuer les actions de bases qui sont souvent récurrentes durant le développement d'un programme tel que l'exécution d'une requête visant à récupérer des données depuis une base et la gestion de leur affichage.

Le principal inconvénient d'un framework est la courbe d'apprentissage qui est plus élevée, car pour maîtriser un framework, une période de formation doit être prise en compte.

4.1.2.1 PHP

Le PHP (Hypertext Preprocessor) est un langage de scripts Open Source, essentiellement utilisé pour la production de pages Web dynamiques à la volée. Il peut être simplement intégré à une page HTML.

Le code est exécuté sur le serveur qui s'occupera de générer les éléments HTML, XHTML ou encore CSS, mais aussi les données telles que les images qui seront ensuite renvoyés au navigateur puis interprétés. Il est également à noter que le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

Le PHP a pour point fort le fait qu'il soit libre et très répandu, possède une grande communauté et est facile d'accès. Cependant, n'étant pas un langage strict, il faut être rigoureux lors du développement notamment en terme de gestion des variables.

Frameworks PHP

Voici une liste des frameworks PHP les plus populaires à l'heure actuelle. L'avantage des frameworks populaire vient du fait que la communauté liée au framework est très active et permet donc de trouver de l'aide rapidement.

— CakePHP 2
— Code Igniter
— Symfony2
— Yii
— Zend Framework 2

L'ensemble de ces frameworks prennent en compte nativement les éléments suivants :

— Le modèle MVC : modèle vue contrôleur, permet de structurer l'application en séparant la partie présentation (vue), la partie base de données (modèle) et la partie applicative (contrôleur).
— Une gestion de templates (gabarits) : La gestion de gabarit permet de distinguer la présentation du code applicatif .
— Une gestion du cache : permet de stocker des pages afin d'optimiser leur temps de chargement. .
— Une gestion des SGBDR(Système de gestion de bases de données relationnelles : qui permet de gérer plusieurs type de base de données telles que MySQL , PostgreSQL ou encore Oracle selon les besoins.
— L'intégration de l'ORM : Mapping de relation objet qui permet la gestion de la base de données sous forme d'objets.
— L'intégration du Scaffolding (échafaudage) : Permet la création d'un espace d'administration de l'application sans aucun développement, uniquement à partir de l'ORM.
— Utilisation de conventions : Oblige les développeurs à utiliser les mêmes conventions de codage afin d'avoir un code uniforme.

4.1.2.2 Java

Java est un langage de programmation orientée objet sous licence GNU GPL, mis au point par la firme Sun Microsystems. Il a été initialement conçu pour être intégré dans les appareils électroménagers, afin de pouvoir les contrôler, les rendre interactifs, mais surtout de permettre une communication entre eux. Ce qui en fait un de ses points forts : la portabilité. En effet, les programmes écrits en Java sont portables sur plusieurs systèmes d'exploitation, tels que UNIX, Windows, Mac OS ou GNU/Linux avec peu ou pas de modifications.

Il est possible de développer des applications client-serveur en Java. Du côté client il existe les applets, ce sont des applications Java qui fonctionnent via un navigateur web, grâce à une machine virtuelle Java. Celles-ci sont utilisées pour fournir des fonctionnalités interactives qui ne peuvent pas être fournis par le langage HTML.

Du côté serveur nous avons les servlets, qui reçoivent les requêtes du navigateur client, effectuent des traitements et lui renvoient les résultats. Elles sont principalement utilisées dans la génération de pages html dynamique utilisant le protocole de communication HTTP.

Java a pour avantages d'être libre, multi-plateforme et possède un grand nombre de librairies disponibles.

Frameworks Java
Nous avons relevé quelques frameworks populaire pour Java :

- Spring 3 MVC Framework basé sur l'architecture MVC sous licence Apache
- JSF (Java Server Faces) Framework basé sur l'architecture MVC
- Apache Struts 2
  Framework sous licence Apache 2.0 également basé sur l'architecture MVC 2 et utilise les servlets, les JSP

4.1.2.3 ASP.NET

ASP.NET est un ensemble de technologies développées par Microsoft permettant la programmation d'applications web dynamiques. Il s'agit du successeur de la technologie Active Server Pages [6] et fait partie de la plateforme .NET de Microsoft (que je décrirais plus loin dans ce rapport). Les pages ASP.NET sont compilées et exécutées par le Common Language Runtime [7]. Cet ensemble de technologies permet de développer des applications web de manière rapide tout en leur fournissant une certaine robustesse. Il existe trois modèles de programmation en ASP.NET :

Le modèle ASP.NET WebForms,
est un des trois modèles de programmation pour créer des sites et des applications web en ASP.NET. Il s'agit d'éléments d'interface qui peuvent être positionnés et qui offrent la possibilité à l'utilisateur de modeler lui-même l'apparence des différentes pages de son application web via une interface de type glisser-déposer.

Ce modèle a pour avantage de fournir une vaste palette de contrôle permettant le prototypage rapide d'une application. Cependant, il n'est pas possible de contrôler directement le code HTML généré en sortie.

Le modèle ASP.NET MVC,
qui, comme son nom l'indique, permet de créer des applications web basées sur le MVC (Model View Controller) : Le modèle (model) décrit les données manipulées par l'application telles que les interactions avec la base de données et est responsable de leur intégrité. La vue représente la partie de l'application qui dirige l'affichage des données récupérées depuis le modèle. Enfin, le contrôleur est la partie qui s'occupe de la gestion des interactions de l'utilisateur avec l'application. En général, le contrôleur lit les données affichées depuis la vue, récupère les interactions de l'utilisateur et envoie l'ensemble des données au modèle.

L'avantage principal de ce modèle est cette séparation de la vue, du modèle et du contrôleur rendant le code plus facile à maintenir. De plus, contrairement au modèle WebForms, le développeur à un contrôle total sur le code HTML rendu. En contrepartie, le temps de développement d'une application est plus élevé.

6. Active Server Pages (ASP) : Suite de logiciels destinée à créer des sites web dynamiques, ASP est une structure composée d'objets accessibles par deux langages principaux : le VBScript et le JScript.

7. Common Language Runtime (CLR) : Nom du composant de machine virtuelle du framework .NET. Il s'agit de l'implémentation par Microsoft du standard Common Language Infrastructure (CLI) qui définit l'environnement d'exécution des codes de programmes.

4.1.2.4 Ruby

Le Ruby est un langage de programmation orienté objet. Il s'agit d'un langage interprété disposant d'un système de typage dynamique ainsi que d'un ramasse-miettes [8] qui est chargé de libérer automatiquement la mémoire, sa syntaxe à pour particularité d'être concise, permettant de rendre le code plus lisible. La philosophie de conception du Ruby considère que toute donnée est un objet, y compris les types, toute fonction est une méthode et que toute variable est une référence à un objet.

Frameworks Ruby
— Ruby On Rails
— Sinatra

4.1.2.5 Python

Python est un langage de programmation haut niveau dont la philosophie de conception mets l'accent (tout comme le Ruby) sur la lisibilité du code. Une de ses particularités est qu'il peut être utilisé en tant que langage de script, mais également en langage objet. Tout comme le JavaScript, il inclut un système typage dynamique ainsi qu'une gestion automatique de la mémoire.

La création d'application web en python passe par le Common Gateway Interface (CGI). Il s'agit d'une passerelle permettant de communiquer avec d'autre programmes situés sur le même serveur.
Un des points fort de Python est que le code source est clair et concis, car il est dépourvu de symbole indiquant le début ou la fin d'une méthode par exemple. Cependant, une configuration du serveur est nécessaire pour l'exécution des scripts.

Frameworks Python
— Django
— Grok
— Pylons

4.1.3 Langage de programmation sélectionné

Pour départager l'ensemble de ses technologies et dans un souci de fournir des résultats dans un délai court, il m'a fallu décider vers quel langage de programmation m'orienter.

Tout d'abord, j'ai vérifié quelles sont les contraintes émises par le client.
« Développement des interfaces graphiques en mode Web à l'aide un langage de programmation différent du langage JAVA. » Ce qui restreint le choix du langage à sélectionner. Mais aussi : *« Mise en place d'une base de données sous Microsoft SQL Server (2008 R2 ou 2012). »*et « Système d'exploitation sous Windows Server 2008 R2 ou Windows Server 2012 ou Windows Server 2012 R2. ». L'utilisation d'un langage de programmation conçu par Microsoft permettrait de profiter des optimisations liées

8. Ramasse-miette : Système de gestion automatique de la mémoire. Il est responsable du recyclage de la mémoire préalablement allouée puis inutilisée.

à l'environnement Windows, mais aussi d'avoir moins d'installations et de configuration à effectuer pour mettre en place l'environnement de travail et ainsi commencer directement la partie programmation de l'application web. Après concertation avec mes encadrants, notre choix s'est alors orienté vers ASP.NET.

Il fallait ensuite décider quel langage utiliser parmi ceux disponible dans les technologies .NET, où les plus populaires sont le C# et le Visual Basic .NET (VB.NET). Ayant pratiqué pendant deux ans le VB.NET pour le développement d'applications Windows pendant mon cursus scolaire, je me suis orienté vers ce langage, car l'idée de pouvoir utiliser ses acquis pour le développement web était très intéressante : cela me permettra à la fois de découvrir l'ASP.NET et de m'occuper de la partie traitement avec un langage que je connaissais déjà. De plus, l'entreprise possède des compétences internes sur ce langage.

Après la sélection du langage, il restait à déterminer quel modèle de programmation adopter parmi WebForms, MVC et Web Page pour la réalisation du projet. Le modèle WebForms a semblé pour moi, comme la solution la plus adéquate pour le développement, car cela ressemblait beaucoup au WinForms [9] utilisés dans la réalisation d'applications Windows. En effet, celle-ci permet de manière intuitive de concevoir la partie Front-End (ce que l'utilisateur verra afficher sur son écran) et de concevoir une maquette des différents visuels de l'application. Ce facteur rapidité de conception de l'interface entre en adéquation avec la démarche agile choisie pour le développement du projet.

4.2 Outils et technologies utilisés

Ayant choisi de me porter sur de l'ASP.NET, l'outil le plus adéquat pour réaliser l'application et ses différentes fonctionnalités, est l'environnement de développement Microsoft Visual Studio 2013. L'utilisation de jQuery et jQueryUI, deux frameworks JavaScript m'a permis de mettre en place un système de boites de dialogue utilisé pour les différents formulaires présents dans l'application. Le suivi des éléments de recette a été appuyé par le logiciel M-Files qui permet d'organiser des éléments en fonction de leurs méta-données.

4.2.1 Environnement de développement

Microsoft Visual Studio est un environnement de développement intégré (IDE) conçu comme son nom l'indique, par Microsoft. Il est utilisé pour le développement de programme informatique pour les systèmes d'exploitations Microsoft Windows , mais aussi pour le développement de site web, d'applications web et services web. Il peut produire aussi bien du code natif [10] que du code managé [11]. Visual Studio inclus un éditeur de

9. Windows Forms (WinForms) : Il s'agit du nom de l'interface graphique incluse dans le framework .NET, qui permet de faire des applications Windows composée de fenêtres.

10. Code natif : Programme dont le code source est composé d'instructions directement reconnues par un processeur.

11. Code managé : Programme dont le code source s'exécute par l'intermédiaire d'une machine virtuelle.

code supportant IntelliSense [12]. D'autres outils y sont inclus tel qu'un outil de création d'interfaces graphiques, un outil de création de classe et de modèle de base de données. Visual Studio supporte divers langages de programmation tels que : le C++, le VB.NET, C#, le F#, le Python, ou encore le Ruby. Il propose également le support du XML/XSLT, du HTML/XHTML, du JavaScript ou encore du CSS.

Nous avons donc décidé que cet outil était plus que nécessaire si l'on envisage le développement de l'application en ASP.Net. En effet, celui-ci permet d'exploiter tout le potentiel des technologies .NET.

Pour la partie visuelle des pages, j'ai choisi d'appuyer les différents contrôles de saisie (dans la mesure du possible) sur du JavaScript, car étant du côté client, la vérification des données saisies est plus rapide que s'il s'agissait d'une vérification du côté serveur. Je suis également passé par jQueryUI [13] pour des fonctionnalités telles que l'affichage de fenêtre flottantes à l'écran, essentiellement utilisées pour afficher les différents formulaires que propose l'application.

4.2.2 Système de gestion de base de données

La solution développée utilise comme système de gestion de base de données (SGBD) Microsoft SQL Server 2012. Comme tout autre (SGBD), il permet le stockage et la gestion des données. Il a pour particularité d'être un SGBD supportant nativement des requêtes SQL qui impliquent plusieurs bases de données.

Microsoft SQL Server utilise le langage T-SQL (Transact-SQL) pour ses requêtes. Il s'agit d'une implémentation de SQL prenant en charge les procédures stockées [14] et les déclencheurs [15]. Il possède également une composante transactionnelle qui lui permet de préparer des modifications sur les données d'une base et de les valider ou de les annuler de façon atomique. Cela permet de garantir la cohérence et l'intégrité des informations stockées dans la base de données.

4.2.3 Suivi des éléments de recette

M-Files est une solution de gestion de contenu (ECM) [16], développé par l'entreprise du même nom, fournissant aux utilisateurs un système d'organisation de contenu basé sur les méta-données qui catégorise les éléments pour ce qu'ils sont et non pas par leur emplacement sur le disque dur.

12. IntelliSense : Terme général employé pour plusieurs fonctionnalités en autocomplétion, aidant l'utilisateur à obtenir des informations sur le code qu'il utilise selon le contexte.

13. jQueryUI : jQuery UI est une collection de composants d'interface graphique et d'effets visuels implémentée avec jQuery (une bibliothèque écrite en JavaScript), du CSS et du HTML.

14. Procédure stockée : Ensemble d'instructions SQL précompilées, stockées dans une base de données et exécutées sur demande par le SGBD qui manipule la base de données.

15. Déclencheurs : Dispositif logiciel qui provoque un traitement particulier en fonction d'évènements prédéfinis.

16. Gestion de contenu d'entreprise :Programme visant à gérer l'ensemble des contenus d'une organisation. Il prend en compte sous forme électronique les informations qui ne sont pas structurées, comme les documents électroniques, par opposition à celles déjà structurées dans les bases de données

M-Files fonctionne sous Windows et peut communiquer avec d'autres programmes via son API fourni. Il supporte tous types de documents électroniques mais également les informations définies par les utilisateurs n'étant pas associées à un document (telles que les contacts, les projets, les clients etc.).

4.3 Une application ASP.NET WebForms

Dans cette partie, je vais exposer les différentes composantes d'une application ASP.NET de type WebForms. En effet, pour pourvoir développer convenablement il est impératif de prendre connaissance du principe de fonctionnement d'une application ASP.NET WebForms.

4.3.1 Framework .NET

Le framework .NET est un framework créé par Microsoft en 2002, étant principalement dédié à la réalisation d'applications s'exécutant dans des environnements Microsoft. Il offre la possibilité de réaliser des programmes fonctionnant sous Windows, ou des sites web ou encore des applications pour appareils mobile. Celui-ci s'appuie sur la norme Common Language Infrastructure [17] (CLI) qui ne dépend pas du langage de programmation utilisé. Ainsi tous les langages compatibles respectant la norme CLI ont accès à toutes les bibliothèques disponibles dans l'environnement d'exécution. Ce framework a été conçu dans le but de faciliter le travail des développeurs en unifiant l'approche de conception d'applications Windows.

L'environnement d'exécution du framework se compose d'un moteur d'exécution, appelé Common Language Runtime (CLR) : il s'agit d'une implémentation de la norme CLI par Microsoft permettant la compilation du code source de l'application en un langage intermédiaire, nommé Microsoft Intermediate Language (MSIL) et se comporte comme une la machine virtuelle Java. Lors de la première exécution de l'application, le code MSIL est à son tour compilé à la volée en code spécifique au système grâce à un compilateur Just In Time [18](JIT). L'environnement d'exécution se distingue en deux parties : une partie liée à l'exécution d'applications Windows natives appelée WinForms et une autre partie relative à l'exécution d'application et de service Web appelé ASP.NET. La figure n° 7 illustre les différents composants du framework.

17. Common Language Infrastructure : Il s'agit d'une spécification ouverte développée par Microsoft pour sa plateforme .NET qui décrit l'environnement d'exécution de la machine virtuelle basé sur Common Intermediate Language (CIL), un langage de programmation bas niveau.

18. Just in time : technique de compilation consistant à traduire le code à octets en code machine au moment de l'exécution d'un programme.

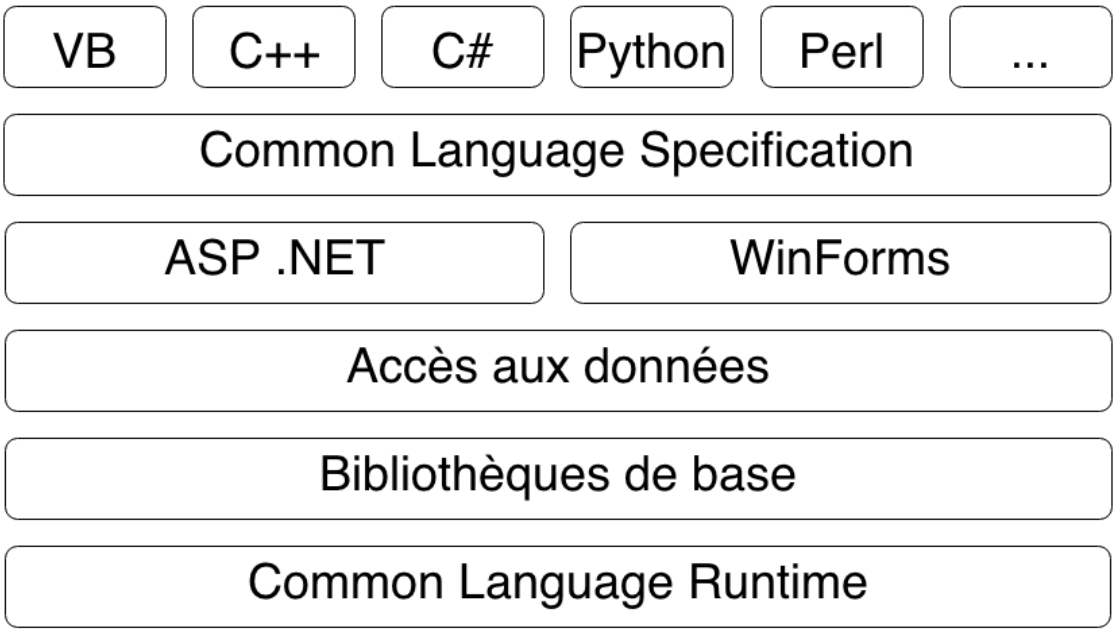


FIGURE 7 – Schéma de l'architecture du framework .NET

4.3.2 Serveur IIS

Les applications ASP.NET sont gérées par *Internet Information Services* (IIS). Il s'agit d'un serveur Web développé par Microsoft qui fournit une plateforme sécurisée et facile à administrer pour l'hébergement des Services Web ainsi que des applications Web. Dans sa version plus récente, IIS permet la gestion de la plupart des langages utilisés sur le Web allant de l'ASP.NET au PHP.

Depuis la version 6.0 de IIS (version actuelle : 8.0), une nouvelle fonctionnalité a été intégrée : le pool d'application. Celui-ci peut contenir une ou plusieurs applications web et offre la possibilité de configurer le niveau d'isolation entre ces différentes applications. Cette fonctionnalité offre la possibilité de :

— grouper les sites et applications qui s'exécutent avec les mêmes paramètres de configuration
— isoler les sites et applications web qui s'exécutent avec des paramètre de configuration unique
— prévenir que les ressources d'une application accèdent aux ressources d'une autre

4.3.3 Architecture d'une application ASP.NET

Une application ASP.NET est exécutée via une séquence de requêtes HTTP et de réponses HTTP. Le navigateur client demande une page ASPX et le serveur web exécute la page et produit le code XHTML, CSS et JavaScript correspondant.

4.3.3.1 Cycle de vie d'une page ASP.NET WebForms

Lorsqu'une page ASP.NET s'exécute, elle passe par plusieurs étapes qui constituent son cycle de vie. Celles-ci sont l'initialisation, l'instanciation des contrôles, la restauration et le maintien de l'état, l'exécution du code relatif à la gestion des évènements et enfin, le rendu. Connaître le fonctionnement du cycle de vie d'une page ASP.NET est important, car cela permet de savoir exactement où placer le code pour qu'il fonctionne correctement. En règle générale, une page ASP.NET passe par les étapes présentées sur la figure n° 8.

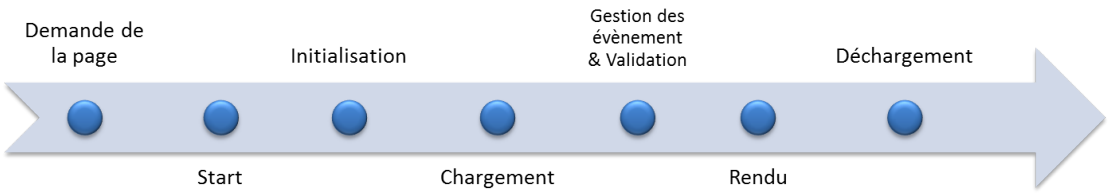


FIGURE 8 – Cycle de vie d'une page ASP.NET WebForms

Demande de la page
La demande page s'effectue avant le début du cycle de vie. Lorsqu'elle est demandée par le client, le moteur ASP.NET détermine si la page doit être analysée et compilée et donc, commencer son cycle de vie, ou s'il s'agit d'une version de la page disponible en cache pouvant être envoyée sans l'exécuter.

Start
Dans cette étape, les propriétés de la page telles que *Request* qui récupère un objet *HttpRequest* permettant à ASP.NET de lire les valeurs HTTP envoyées par le client pour la page demandée et *Response* (qui récupère l'objet *HttpResponse* permettant d'envoyer la réponse HTTP au client.) sont définies. A cette étape, la page détermine également si la demande est une publication (aussi appelé « postback ») ou une nouvelle demande, la propriété *IsPostBack* est alors définie.

Initialisation de la page
Pendant l'étape d'initialisation, les contrôles de la page sont disponibles et pour chacun d'eux, la propriété *UniqueID* est définie. Si la demande est une publication (*IsPostBack=true*), alors les données et les valeurs des propriétés ne sont pas chargées. S'il s'agit en revanche d'une nouvelle demande, les données et valeurs des contrôles sont récupérées.

Chargement
Pendant le chargement, si la requête est une demande de publication, alors les propriétés des contrôles sont chargées avec les informations récupérées depuis les états d'affichage et de contrôles.

Gestionnaires d'évènement & Validation
Durant cette étape, dans le cas d'une demande de publication, tous les gestionnaires d'évènements sont appelés, suivi de la méthode de validation qui affecte la propriété *IsValid* de chaque contrôle ainsi que de la page.

Rendu
Avant cette étape, l'état d'affichage de la page et des contrôles qu'elle contient est sauvegardé. Pendant l'étape du rendu, la page appelle la méthode *Render* de chaque contrôle, apportant un flux de sortie *OutputStream* qui sera intégré dans la propriété *Response* de la page pour pouvoir être interprété par le navigateur.

Déchargement
Après le rendu complet de la page, celle-ci passe en phase de déchargement. Cette méthode est appelée après que la page ait été rendue dans sa totalité et envoyée au client. A cette étape, les propriétés de la page telles que *Response* et *Request* sont déchargées et tous les nettoyages sont effectués.

4.3.3.2 Gestion des évènements

Une des caractéristiques clé d'une application ASP.NET Webforms est qu'elle utilise un modèle de programmation basé sur les évènements. Ces évènements sont capturés par des gestionnaires d'évènement (event handler) : ce sont des méthodes qui déterminent quelles actions ont été effectuées lorsqu'un évènement se produit, tel que le clic sur un bouton ou encore la sélection d'un élément dans une liste déroulante. Cette pratique a été inspiré de ce qui se fait généralement sur les page Web : une script client qui s'exécute sur l'évènement *onclick* d'un bouton en HTML. ASP.NET apporte donc ce modèle au traiement serveur.

Il est a noté que le système de gestion d'évènements utilisé par ASP.NET ne fonctionne pas de même façon que celui d'une application Windows standard. En effet, dans une application Windows les évènements sont produits et gérés par la même machine, contrairement aux évènements ASP.NET qui sont produit par le navigateur, puis transmis et géré par le serveur.

4.3.3.3 le ViewState

Il existe plusieurs méthodes pour stocker l'information d'un état à l'autre lors d'une navigation sur un site :
— Les variables de session
— Les cookies
— Le stockage en base de données

Bien que ces méthodes soient efficaces, elles possèdent leurs limites. En effet, il n'y a pas de garantie que le navigateur client accepte les cookies pour pouvoir stocker l'information. De plus, les variables de session sont des variables temporaires, les éléments qui y sont conservés ne le sont que jusqu'à la durée de la session. De même, le stockage en Base de Données est coûteux en conception et développement, car il faudrait pour chaque utilisateur par exemple , stocker dans la base les informations relatives à l'état

de la page en cours de visualisation. Le framework .Net propose un concept qui est la clé maîtresse du développement en ASP.NET WebForms : le ViewState.

Le ViewState conserve l'état de chaque élément .NET (qui sont contrôlés du côté serveur), dans la page HTML qui est en cours d'exécution.
De ce fait, par défaut si on place une zone de texte (TextBox) sur une page et qu'on envoie la page à un client, celui-ci recevra les éléments HTML correspondant à la zone de saisie, mais aussi un élément caché (<INPUT TYPE="HIDDEN" .../>) qui stockera l'état de l'objet.

En observant le code source d'une page ASP.NET WebForms, on peut remarquer le champ caché suivant :

```
1  <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="JvXuP8+l6ll+6UwtGGv44/
       uvcDmj[...]" />
```

Cet élément contient à lui seul tous les états des contrôles de toute la page web. Le Viewstate est (par défaut) envoyé au navigateur, puis est retourné au serveur sous la forme d'un champ caché à l'intérieur de la page cela permet de donner l'impression à l'utilisateur que l'application web se souvient de ses actions précédentes sur la page. Cependant, en stockant plus la page contient des contrôles, plus la quantité d'information à sauvegarder devient importante, donc, le Viewstate gagnera du volume ainsi que la page web. Il faut donc l'utiliser avec parcimonie.

4.4 Méthodologie de production

Dans cette partie, je détaillerai le cheminement effectué pour réaliser les pages de l'application web, ainsi que leurs liaisons à la base de données.

4.4.1 Ordre de développement des écrans de l'application

Chaque page de l'application représente une ou plusieurs fonctionnalités où certaines dépendent d'autres en termes de données. Nous avons donc décidé de développer dans un premier temps les vues qui concernent les données auxiliaires. Cette démarche à pour but de rendre disponibles toutes les données annexes lors du développement des fonctionnalités principales de l'application : ces données apparaîtront dans des listes déroulantes et aussi pour que les clés étrangères soient correctement complétées lors de l'insertion ou la modification de l'enregistrement dépendant de ces clés dans la base de données. La dépendance des fonctionnalités est illustrée par la figure n° 9.

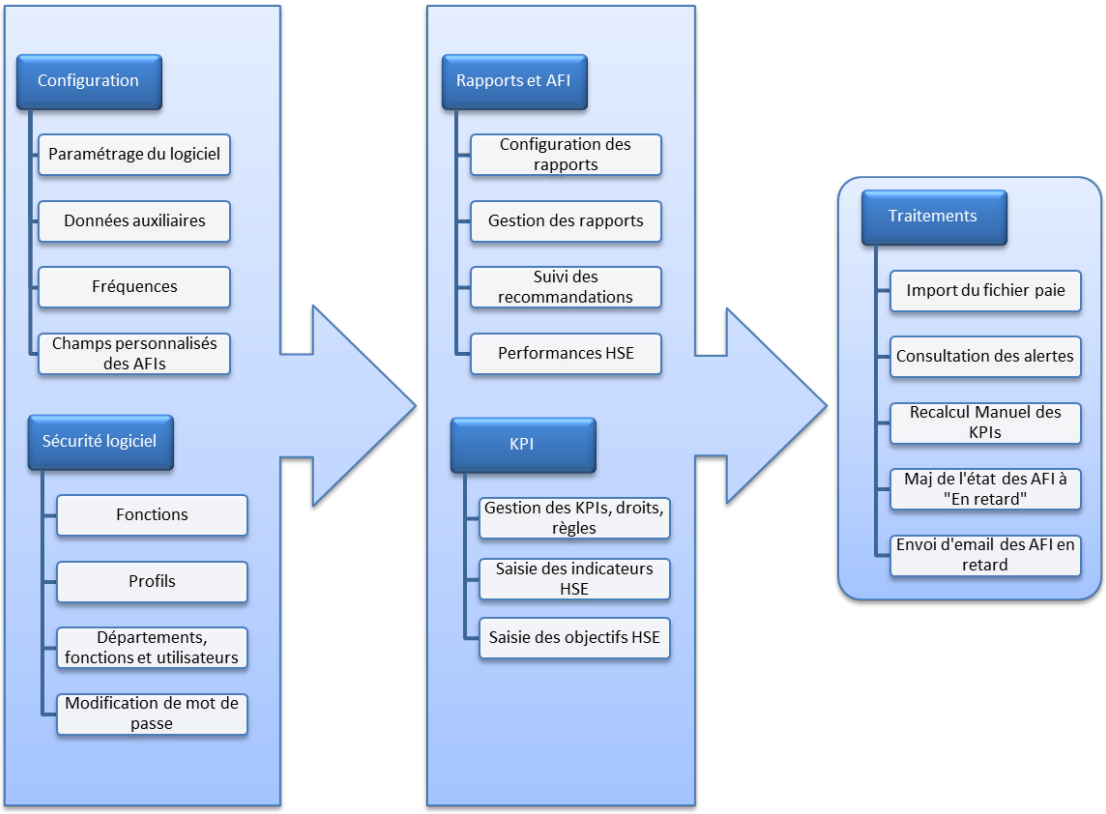


FIGURE 9 – Dépendance des fonctionnalités de l'application

Dans un premier temps, je me suis penché sur la création des pages *Données auxiliaires* et *Fréquences* car les données données gérées par ces écrans sont utilisées dans la plupart des pages de l'application. J'ai ensuite procédé à la création des pages *Fonctions* et *Profils* qui permettent la personnalisation du menu de l'application en fonction des droits de l'utilisateur.

Ces dernières pages m'ont permises par la suite de réaliser les fonctionnalités *Départements, fonctions et utilisateurs* et *Modification de mot de passe* qui permet comme son nom l'indique la gestion des données relatives aux départements, aux fonctions qu'ils contiennent et aux utilisateurs associés. Ceci étant fait, j'ai pu mettre en place le système d'authentification qui s'affiche à l'écran d'accueil de l'application et procédé à l'écran de paramétrage du logiciel, qui permet la définition de plusieurs paramètre globaux de l'application (tels que la configuration de la complexité du mot de passe utilisateur par exemple).

Une fois tout cela en place, j'ai procédé au développement des pages relatives aux indicateurs de performances et à leur saisies *Gestion des KPIs, droits, règles* , *Saisie des indicateurs* et *Saisie des objectifs* . Pendant la réalisation de ces écrans , Anthony Picard, un stagiaire de l'école informatique SUPINFO s'est occupé des pages liées à la gestion des rapports de l'entreprise cliente : *Configuration des rapports* et *Gestion des rapports*.

Enfin, je suis passé à la réalisation des derniers écrans de l'application qui sont : *Suivi des recommandations*, qui se base sur les données des derniers écrans, *Performance HSE* dépendant des données relatives aux indicateurs et aux recommandations et *Consultation des alertes* qui recensera l'ensemble des anomalies liées aux traitements.

Après avoir terminé le développement de ces fonctionnalités, je suis ensuite passé à la réalisation des traitements clés de l'application dans plusieurs procédures stockées en T-SQL [19]. Cette partie s'effectue sur la base de données, car dans un soucis d'évolution de l'application, ces traitements ne seront pas affectés si elle subit des modifications.

Note : le développement de toutes ces fonctionnalités a été réalisé en parallèle avec le traitement des différents éléments de recettes postés par le client.

4.4.2 Méthode de développement pour une page ASP.NET

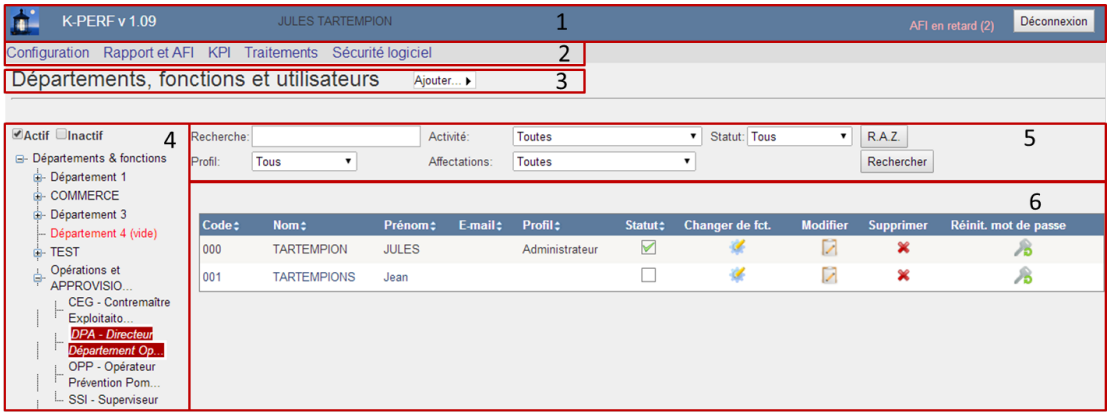
L'ensemble des fonctionnalités de l'application ont suivi le processus de développement suivant (figure n° 10) :

FIGURE 10 – Cheminement du développement d'une page ASP.NET

Réalisation de l'interface graphique

Tout d'abord, je positionne les différents éléments WebForms et HTML sur la page, que ce soit via le concepteur de vue ou le code source. Puis, je crée les formulaires d'ajout, de modifications ainsi que les autres fenêtres jQueryUI : ces éléments sont encadrés par des balises <*div*> avec un identifiant précis commençant généralement par *dialog-* . La figure n° 11 expose les différents composants d'un écran type de l'application web.

19. T-SQL : Extension propriétaire du langage SQL, destiné à l'accès aux bases de données Microsoft.



Légende

1: Bandeau de connexion

2: Menu dynamique

3: Titre de la page

4: Arborescence modifiable

5: Zone de filtres

6: Affichage des données

FIGURE 11 – Ecran type de l'application web

Une boite de dialogue de jQueryUI est gérée par deux fonctions (présentées ci-dessous), l'une permet de l'initialiser, l'autre permet de l'afficher, avec des traitements annexes si nécessaire (tels que l'affichage ou dissimulation d'éléments du DOM dans le formulaire). Une fois que tout est en place, je passe à l'étape suivante.

```
1  $(function () {
2  //Definition des parametres d'initialisation de la boite de dialogue jQueryUI
3            $("#dialog-login").dialog({
4                title: "Connexion",
5                draggable: false,
6                resizable: false,
7                width: 500,
8                modal: true,
9                autoOpen: false
10
11           });
12       });
13
14   //Fonction affichant la div 'dialog_login' en tant que boite de dialogue jQueryUI
15 function showDialogLogin() {
16
17     $("#dialog-login").dialog("open");
18 }
```

Je passe enfin à la création des autres fonctions de vérification des champs (telles que les contrôles de saisie utilisateur par exemple)

Liaison des éléments WebForms à la base de données

Une fois les fonctions JavaScript créées, je passe ensuite à la liaison des objets WebForms avec la base de données. Celle-ci s'effectue généralement grâce à l'élément *SqlDataSource* qui permet de renseigner directement la requête de récupération des données et de les afficher dans une liste déroulante (*DropDownList*) ou encore un tableau

(*GridView*) en y précisant l'identifiant du *SqlDataSource* concerné dans la propriété *DataSource*. Le code ci-dessous montre un exemple de la liaison de données sur une liste déroulante :

```
1  <asp:DropDownList ID="DropDownListLogin" runat="server"
2   DataSourceID="SqlDataSourceUsers" DataTextField="Utilisateurs"
3     DataValueField="CODE">
4  </asp:DropDownList>
5
6  <asp:SqlDataSource ID="SqlDataSourceUsers" runat="server"
7   ConnectionString="<%$ ConnectionStrings:KPIDEVConnectionString %>"
8   SelectCommand="SELECT NOM + ' ' + PRENOM + ' (' + CAST(CODE AS VARCHAR) + ')' AS
                   Utilisateurs, CODE FROM SYS_USER ORDER BY NOM, PRENOM">
9   </asp:SqlDataSource>
```

S'ensuit alors de la configuration de l'affichage des données qui peut également s'effectuer depuis le concepteur de vue ou directement dans le code source de la page.aspx.

Programmation évènementielle

Après avoir procédé à la liaison des données, je m'occupe enfin de la partie programmation en VB .NET de la page, celle-ci est basée sur les évènements des différents éléments WebForms, mais aussi sur le cycle de vie de la page. Par exemple, le code dans la procédure ci-dessous s'exécutera lors du chargement de la page :

```
1  Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me
       .Load
2          If Not IsPostBack Then
3              getMenu()
4          End If
5  End Sub
```

Je passe ensuite au développement des fonctionnalités nécessaires à la page qui sont généralement l'ajout, la modification ou la suppression de données tout en prenant en compte les différentes règles de gestion établies dans le document de spécifications techniques et fonctionnelles. Pour chaque fonctionnalité réalisée j'effectue des tests et applique les corrections nécessaires en cas de dysfonctionnement.

4.4.3 Traitement des indicateurs

Une fois le développement des différentes fonctionnalités effectué, j'ai procédé par la suite à la création des différentes procédures stockées qui se chargeront de calculer les indicateurs de performance selon l'algorithme fournit par le client, celui-ci se présente sous la forme :

```
Itération de tous les indicateurs actifs d'une nature et d'une origine précise
    Itération des mois faisant partie de la fréquence de l'indicateur
        Méthode de calcul ?
            Incrément
                Règle de calcul comporte une condition ?
                    Oui
                        La condition est respectée ?
                            Oui
                                Valeur de l'indicateur = Valeur de l'indicateur + 1
                            Non
                                Aucune action à effectuer
                    Non
                        Valeur de l'indicateur = Valeur de l'indicateur + 1
            Somme
                Règle de calcul comporte une condition ?
                    Oui
                        La condition est respectée ?
                            Oui
                                Valeur de l'indicateur = Valeur de l'indicateur +
                                    valeur du champ numérique
                            Non
                                Aucune action à effectuer
                    Non
                        Valeur de l'indicateur = Valeur de l'indicateur + valeur du
                            champ numérique
```

Tous les indicateurs de performance possèdent une nature et une origine, l'algorithme précédent s'appliquera à l'ensemble des indicateurs d'une origine en particulier, à quelques différences près. De plus, chaque indicateur possède une fréquence de type mensuelle, bimensuelle, trimestrielle, semestrielle ou encore annuelle (personnalisable dans l'application). En effet, si la date du lancement du traitement ne correspond pas à la fréquence définie lors de la création de l'indicateur, alors le traitement n'aura pas lieu. Un indicateur possède également une méthode de calcul avec la possibilité d'avoir une règle de calcul définie par l'utilisateur dans l'écran de gestion des indicateurs. S'il possède une règle, celle-ci doit être respectée pour que le calcul puisse s'effectuer.

4.4.4 Déploiement des itérations

Chaque itération de l'application est déployée sur un serveur IIS situé à run [in] box où le client dispose d'un accès pour effectuer ses tests. Ces itérations contiennent généralement des fonctionnalités supplémentaires ainsi qu'un bon nombre d'éléments de recette traités. Il ne lui reste plus qu'à recetter ou non les différents éléments traités et d'en déclarer de nouveaux si nécessaire, qui seront traités pour les itérations à venir.

4.5 Difficultés rencontrées

4.5.1 Problème d'exécution de script côté serveur

Une des principales difficultés rencontrées lors du développement a été l'utilisation du contrôle Web Forms UpdatePanel à l'intérieur d'une fenêtre jQueryUI qui est affichée depuis le code VB.NET. En effet, le contrôle UpdatePanel fait partie des fonctionnalités incluses dans Microsoft ASP.NET 3.5 AJAX Extensions, qui permet le rendu partiel de pages sans exiger de rafraîchissement complet de la page. Son utilisation nécessite la mise en place d'un contrôle nommé ScriptManager.

Le problème lié avec ce contrôle est que toutes les fonctions JavaScript qui étaient appelées depuis le serveur ne se lançaient plus. En effet, le code utilisé avant l'utilisation du UpdatePanel était le suivant :

```
'Affichage de la pop-up jQueryUI au prochain chargement de la page
Dim sb As System.Text.StringBuilder = New System.Text.StringBuilder()

sb.Append("<script language='javascript'>")

sb.Append("$('#dialog-ajout').attr('title', 'Modifier une donnée');")

sb.Append("$(document).ready(function() {$('#dialog-
ajout').dialog('open');});")

sb.Append("</script>")
If (Not ClientScript.IsStartupScriptRegistered("JSScript")) Then

   ClientScript.RegisterStartupScript(Me.GetType(), "JSScript", sb.ToString())

End If
```

Le problème se situait au niveau de la condition encadrée ci-dessus. En effet l'objet ClientScript n'était plus pris en compte depuis la mise en place du ScriptManager, qui est configuré pour le rendu partiel de la page et n'exécutera pas le code , car pour pouvoir s'exécuter, la fonction d'affichage de la boîte de dialogue nécessite le rafraîchissement de la page pour être appelée par le serveur. Il a donc fallu trouver une alternative à ce problème.

Après avoir consulté la documentation fournie par Microsoft, il s'est avéré que l'objet ScriptManager possédait également une méthode RegisterStartupScript fonctionnant pour des communications client-serveur asynchrones, permettant ainsi d'appeler le code JavaScript entré en paramètre, au début de la prochaine publication. Il suffisait ensuite de déterminer s'il s'agissait d'une communication asynchrone ou non.

```
Public Shared Sub RunJavascript(ByVal _
   oScriptManager As Web.UI.ScriptManager, ByVal JScode As String)

        If oScriptManager.IsInAsyncPostBack Then

            oScriptManager.RegisterStartupScript(oScriptManager.Page, _
            oScriptManager.Page.GetType, "JSCode" & Rnd(1).ToString, _
            JScode, True)

        Else

            oScriptManager.Page.ClientScript.RegisterStartupScript( _
            oScriptManager.Page.GetType, "JSCode" & Rnd(1).ToString,
            JScode, True)

        End If

End Sub
```

La procédure ci-dessus prend en compte deux paramètres : l'objet ScriptManager de la page concernée et le code JavaScript formaté en chaîne de caractères. La procédure vérifie s'il s'agit d'une publication (postback) asynchrone grâce à la propriété IsInAsyncPostBack et si c'est le cas, elle appelle la méthode RegisterStartupScript() du script manager, sinon elle appelle la méthode du même nom de l'objet client script RegisterStartupScript.

4.5.2 Problème de publication avec des boutons ASP.NET dans une fenêtre jQueryUI

Une autre difficulté rencontrée a été l'impossibilité de déclencher les évènements associés aux boutons WebForms lorsqu'ils étaient placés à l'intérieur d'une boîte de dialogue jQueryUI. En effet, pour plus d'aisance dans le développement, ces boutons permettent de déclencher directement le traitement du serveur, au lieu de passer par du JavaScript et de lui indiquer quel évènement est à déclencher. Pour comprendre ce qui ne fonctionnait pas , il fallait observer le comportement d'une boîte de dialogue jQueryUI.

Une page web ASP.NET contient la balise HTML *<form>* qui est considérée comme un formulaire de la page en HTML standard. De plus, les boutons WebForms sont rendus sous la forme de *<input type="submit" ../>* qui soumettent les informations du formulaire au serveur.

Lors de l'affichage d'une de ces boites de dialogue, bien qu'elle reprenne le contenu de l'élément *div* associé dans la page HTML, celle-ci ne fait plus partie du formulaire, ce qui créé un dysfonctionnement au niveau des boutons de soumission.

Pour la résolution de ce problème, il fallait indiquer à la fenêtre de s'intégrer au formulaire de la page HTML. Pour ce faire, il fallait ajouter le code ci-dessous dans les paramètre d'initialisation de la fenêtre jQueryUI, plus précisément au moment de son affichage.

```
1  $(this).parent().appendTo("form");
```

Une fois placé dans le paramètres d'ouverture de la fenêtre jQueryUI on obtient alors le code suivant :

```
1  $(function () {
2  //Définition des paramètres d'initialisation de la boite de dialogue jQueryUI
3            $("#dialog-login").dialog({
4
5                 //Intégation de la fenêtre au formulaire de la page
6                 open: function (type, data) { $(this).parent().appendTo("form");},
7
8                 title: "Connexion",
9                 draggable: false,
10                resizable: false,
11                width: 500,
12                modal: true,
13                autoOpen: false
14
15           });
16       });
```

4.5.3 Problème de tri

Un problème a été remonté par le client indiquant que le tri du code identifiant des données auxiliaires n'était pas trié correctement. En effet le champ textitCODE dans la base de données est de type *varchar*[20] et certains codes sont composés de caractères numériques et d'autres de caractères alphabétiques.

La page des données auxiliaires est conçue de telle sorte que pour une seule page, le paramètre dans son URL permet de définir quelles données de quelle table feront l'objet d'un affichage.

Par exemple la figure n° 12 affiche toutes les données issues de la table *UNITE*, et en cliquant sur les éléments de l'arborescence, une requête de récupération se chargera de rapatrier les données de la table concernée dans le tableau. La structure de données dans ces tables étant la même, le champ *CODE* peut donc contenir, selon les tables, des caractères numériques ou alphabétiques.

20. varchar : Chaîne de caractères

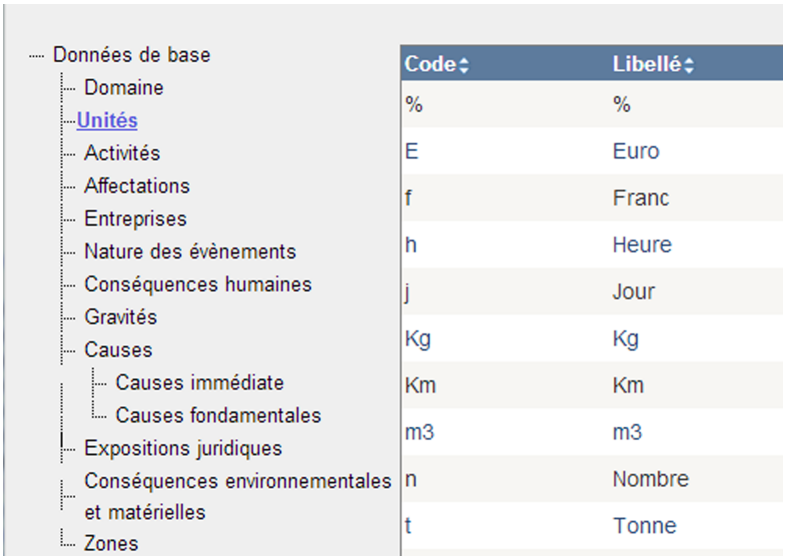


FIGURE 12 – Aperçu de l'écran des données auxiliaires

Or, le tri s'effectuant par la commande SQL *ORDER BY*, il est normal de trouver une séquence de code allant de 1 à 20 triée de la manière suivante :

```
[CODE]
1
10
11
12
(...)
19
2
20
3
(...)
9
```

Il fallait donc trouver comment satisfaire à la fois un tri d'entiers et un tri de chaînes de caractère en une seule requête SQL afin de préserver l'aspect évolutif de la page.

Pour résoudre cette problématique, j'ai utilisé la fonction *TRY_CONVERT* sur le champ *CODE* dans la requête afin de le convertir en entier. Cette fonction retourne soit le champ converti, soit *NULL*, si la conversion a échoué. Une fois la conversion réalisée, je reconvertie le champ en entier par la même fonction puis j'ajoute à sa gauche une vingts de zéro (correspondant à la taille totale du champ) et j'utilise la fonction *RIGHT* afin de ne prendre que les 20 premiers entiers de droite, dans le but d'ordonner le champ s'il contient des entiers. Le tout est mis en paramètre dans la fonction *ISNULL* qui renvoie une expression à la place d'une autre si celle-ci renvoie *NULL*. En effet, si le premier *TRY_CONVERT* échoue, alors il retournera *NULL* , ce qui génèrera aussi un *NULL* dans le second *TRY_CONVERT* ,qui sera intercepté par la fonction *ISNULL* dans laquelle j'indique de retourner le champ *CODE* tel qu'il est.

Dans l'exemple suivant, nous admettons que le champ CODE ne retourne que des entiers.

Exemple : On concatène le champ avec autant de zero que la taille du champ *CODE* :

```
00000000000000000001
00000000000000000010
00000000000000000011
...
00000000000000000002
```

Puis on récupère les vingts premiers chiffre en partant de la droite :

```
0[00000000000000000001]
00[00000000000000000010]
00[00000000000000000011]
...
0[00000000000000000002]
```

Ainsi, la commande ORDER BY ordonnera les différents entiers retournés.

La requête SQL se présente sous la forme suivante :

```
1 SELECT  ID, CODE,  ISNULL(RIGHT('00000000000000000000'+TRY_CONVERT(NVARCHAR,
      TRY_CONVERT(INTEGER,CODE)),20), CODE) as CODE_FORMAT ,LIBELLE, LIFE_CYCLE
2 FROM NOM_TABLE ORDER BY CODE_FORMAT;
```

La colonne CODE_FORMAT servira de base pour le tri et les données affichées seront issues du champ CODE

5 Conclusion et Perspectives

A la fin du stage, j'estime à 70% le taux de réalisation du projet. L'entreprise poursuivra le développement de l'application en vue de la mise en exploitation prochainement chez le client final.

Bien que le délais dans ce projet était un critère important, au terme de ce stage la réalisation complète du projet n'a pas été atteinte. Essentiellement, malgré l'application du processus agile de développement, du retard a été pris :

— D'une part, par une mauvaise évaluation du temps de développement de certaines fonctionnalités qui se sont avérées plus complexes que prévu
— D'autre part, le retard pris par le manque de disponibilité du client pour la validation et la recette de certaines fonctionnalités (manque de temps ou congés).

En conclusion, bien que je ne pourrai participer à la finalisation du projet, ce stage m'a permis d'acquérir de nombreuses connaissances, notamment dans le domaine du développement d'application web avec le framework .NET via ASP.NET WebForms qui permet la réalisation d'applications évènementielle orientée web et de mettre en pratique ces connaissances. J'ai pu également découvrir la vie au quotidien à l'intérieur d'une SS2I ainsi que ses diverses activités.

Ce stage m'a aussi permis d'assimiler des connaissances dans le déroulement du processus de développement agile mis en place par l'entreprise, qui , bien qu'il y ait eu quelques aspects impondérables engendrant du retard et qu'il soit quelque peu fastidieux à gérer l'aspect développement et gestion des éléments de recette, met le client au centre du développement afin de répondre au mieux à ses besoins.

A l'avenir, je souhaiterai approfondir d'autres aspects du framework .NET tel que le modèle ASP.NET MVC bien qu'il soit encore au même niveau de maturité que le modèle WebForms, commence à se populariser.

6 Bibliographie

6.1 Références bibliographiques

— Thuan Thai et Hoang Lam, .NET Framework Essentials (2003)
— Fielding, Roy T., Gettys, James, Mogul, Jeffrey C., Nielsen, Henrik Frystyk, Masinter, Larry, Leach, Paul J., Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1(June 1999).
— Robert J. Oberg, Peter Thorsteinson, Dana L. Wyatt, Application Development Using Visual Basic and .NET (Juin 2002)

6.2 Webographie

— http ://msdn.microsoft.com/fr-FR/
— http ://www.iis.net/
— http ://www.codeproject.com/Articles/457647/Understanding-ASP-NET-Application-and-Page-Life-Cy
— https ://tools.ietf.org/html/rfc2616
— http ://tools.ietf.org/html/rfc2109
— http ://tools.ietf.org/html/rfc3875

A Annexes

A.1 Techniques de gestion d'état fournies par HTTP

La plupart des applications web utilisent le protocole HTTP [21]. Cependant, selon le RFC [22] 2616, ce protocole est générique et sans état et donc ne peut pas gérer de couches d'informations supplémentaire. En effet, un serveur HTTP traite chaque requête qu'il reçoit de manière totalement indépendante les unes des autres. Celui-ci ne mémorise pas les informations du client entre ses différentes requêtes. Pour remédier à ce problème, il existe différents moyens qui permettent de retenir ses informations tels que l'utilisation de cookies, de champs cachés ou d'insérer des paramètres dans l'URL de la page web. Ceux ci permettent de créer un contexte qui permettra au serveur d'identifier le client.

A.1.1 Les cookies

Un cookie (ou témoin de connexion) est un petit paquet contenant l'information sur l'état que le navigateur et le serveur cible peuvent s'échanger pour maintenir des données relatives à l'utilisateur. Dans sa forme la plus simple,cookie n'est autre qu'une paire nom / valeur. Généralement un cookie contient aussi un bon nombre d'attributs tels que la version, un domaine de validité, un chemin spécifiant l'ensemble des URLs sur lequel il s'applique, ainsi que sa période de validité. Pour comprendre comment un cookie stocke des informations, nous allons suivre son processus de création :

Pour accéder à la page www.monsite.com/index.html, le navigateur se connecte au serveur www.monsite.com et envoie une requête similaire à celle-ci :

```
GET index.html HTTP1.1

Host : www.monsite.com
```

Le serveur répond en envoyant la page demandée, précédée par un texte similaire appelé réponse HTTP. Ce paquet peut contenir des lignes demandant au navigateur de stocker des cookies :

21. HTTP : Hypertext Transfer Protocol, Protocole de communication client-serveur développé pour le World Wide Web.

22. RFC : Requests For Comments , Documents officiels décrivant les aspects techniques d'Internet.

HTTP1.1 200 OK

Content-type : texthtml

Set-Cookie : nom=valeur

(contenu de la page)

Set-Cookie est une requête pour que le navigateur stocke la chaîne nom=valeur et la renvoie dans toutes les futures requêtes au serveur. Les information seront donc incluses dans toutes les autre requêtes faite au même serveur (si le navigateur autorise l'utilisation de cookies).
Par exemple, le navigateur appelle la page www.monsite.com/page.html en envoyant au serveur www.monsite.com la requête suivante :

GET page.html HTTP1.1

Host : www.monsite.com

Cookie : nom=valeur

Accept : **

C'est grâce à l'attribut "Cookie" que le serveur sait que cette requête est reliée à la précédente et que l'information est conservée. Le serveur pourra également y stocker des informations.

A.1.2 Les champs cachés

Le principe de cette méthode est d'inclure un champ caché qui contient l'identifiant de la session actuelle (ou d'autres informations dans d'autres champs cachés), à l'intérieur des formulaires des pages HTML envoyées au navigateur. Ce champ caché retournera l'information au serveur qui pourra identifier la session en cours. L'extrait de code ci-dessous montre un exemple de champ caché dans un formulaire en HTML.

```
1 <form method="post" action="url">
2     <input type="hidden" name="ID\_SESSION" value="123" />
3     ...
4     <input type="submit" />
5 </form>
```

L'inconvénient de cette approche est qu'elle nécessite un effort de programmation fastidieux, car toutes les pages doivent être générés dynamiquement pour inclure ce champ caché. De plus, l'information stockée dans le champ cachée est affichée en clair dans le code source de la page web, si un utilisateur tente d'y accéder, il aura la valeur de chaque information cachée. L'avantage est tous les navigateurs prennent en charge ces champs.

A.1.3 Paramètres d'URL

Le principe de cette technique est d'inclure l'ensemble des paramètres nécessaires, comme un identifiant unique de la session par exemple dans tous les URLs émises par le client. Par exemple dans l'url suivante, l'identifiant de la session est passé en paramètre.

http ://www.monsite.com/page ;idsession=valeur

Pour réaliser cette *technique*, toutes les URLs susceptibles d'être envoyées au client (telles que <a href='url'>ou <form action='url'>par exemple) doivent être formatées pour y inclure les paramètres nécessaires. Là encore l'application de cette approche nécessite un certain effort de programmation. L'avantage est que contrairement aux cookies, tous les navigateurs supportent cette méthode.

A.2 Liste des contrôles WebForms les plus utilisés dans l'application

Nom du contrôle serveur	**Description**	**Évènements couramment utilisés**
Label	Affiche du texte sur la page HTML.	Aucun
TextBox	Zone de texte pour un formulaire HTML.	TextChanged
Button	Un bouton normal utilisé pour déclencher des évènements liés au clic, sur le serveur.	Click
LinkButton	Même fonctionnalité que le contrôle Button mais à l'apparence d'une lien hypertexte.	Click
ImageButton	Peut afficher une image cliquable, communique avec le serveur un fois cliqué fournissant des informations telles que les coordonnées de la souris lors du clic sur l'image.	Click
DropDownList	Une liste déroulante dans laquelle on peut lier des données depuis une source.	SelectedIndexChanged
GridView	Tableau permettant l'affichage de données depuis une source, possède également la possibilité d'ajout, de modification et de suppression de données.	OnRowDataBound, OnRowCommand, OnSorting, OnSelectedIndexChanged
CheckBox	Case à cocher similaire à sa version HTML mais contrôlable par le serveur	CheckChanged
TreeView	pour afficher des données hiérarchiques	TreeNodeCheckChanged, SelectedNodeChanged, TreeNodeDataBound

A.3 Infrastructure détaillée du datacenter de run [in] box