

Teddy ALBINA

Concepteur Développeur Informatique
Nextformation - PARIS



Dossier de synthèse

Conception d'un site e-commerce

Sommaire

Remerciements	3
Introduction.....	4
1/ Présentation de l'entreprise	5
1. Organisation et domaines d'activité de la société	5
2/ Mes missions.....	7
3/ Analyse du projet	8
1. Règles de gestion et processus métier	8
Processus métier	8
Règles de gestion.....	9
4/ Méthodologie.....	10
1. Analyse de l'organisation	10
Identification des acteurs du système d'information	10
Communication au sein de l'organisation et avec l'extérieur	11
2. Utilisations de la solution par l'organisation et les acteurs externe	12
3. Conception de la base de données	15
1. Identification des objets métiers.....	15
2. Identifications des relations	15
3. Fiabilité et monté en charge.....	15
5/ Réalisation du projet.....	16
1. Technologies et outils.....	16
2. Organisation de la solution	17
3. Intégration graphique.....	18
1. Mise en place des templates	23
2. Utilisation des templates.....	24
4. Architecture.....	25
1. Composants	26
2. Mise en œuvre.....	29
5. Intégration du système de vente en ligne.....	31
6. Intégration de composants tiers	36
7. Test et validation des développements	39
8. Déploiement.....	41
1. Choix de l'hébergeur	41
2. Mise en place de la solution	42

3. Configuration de la solution	44
6/ Changements en cours de projet.....	45
Conclusion	46
Bibliographie.....	47
ANNEXE	48

Remerciements

Je remercie Mademoiselle Hülya SOMAR, ainsi que l'ensemble de nos formateurs pour leur soutien et leur accompagnement.

Je remercie également Monsieur Franck MOUTARDIER, mon tuteur pour m'avoir accueilli au sein de son entreprise pour la réalisation de mon année de licence professionnelle. De plus, j'adresse mes remerciements à Monsieur Sylvain GRENIER associé de Monsieur MOUTARDIER de m'avoir accueilli au sein de leur société ROUTE INN' MOTO.

Introduction

Mon année de formation, s'est déroulée au sein de l'entreprise Wyzapps software, qui est une SSII crée et dirigée par Monsieur Franck MOUTARDIER.

Wyzapps software est prestataire de service pour la réalisation d'un projet web, pour le compte de l'entreprise Route Inn' Moto, fondée par Messieurs Sylvain GRENIER et Franck MOURTARDIER. Route Inn' Moto est principalement un importateur qui vend des motos 125cm³ sous sa marque DRIM. De plus Route Inn' Moto, a d'autres activités telles que la réparation de deux roues, la vente/reprise de matériel d'occasion et également la vente de produits dérivés.

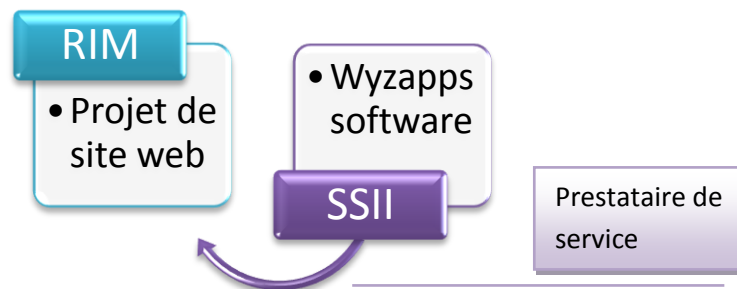
La problématique soulevée par Route Inn' Moto était de posséder un site web, lui permettant de communiquer avec ses clients au travers d'internet, mais aussi de vendre ses produits tant aux professionnels qu'aux particuliers par ce biais.

Dans un premier temps je vous présenterai les deux entreprises dans lesquelles j'ai évolué, puis je vous exposerai les missions qui m'ont été confiées. Par la suite, j'expliquerai comment j'ai procédé de la phase d'analyse en passant par les méthodologies utilisées. Enfin, j'aborderai la réalisation du projet, et les changements d'orientation qui ont eu lieu au cours de la réalisation de ce dernier.

1/ Présentation de l'entreprise

1. Organisation et domaines d'activité de la société

J'ai effectué mon année au sein de la SSII Wyzapps software en tant que prestataire de service pour l'entreprise ROUTE INN' MOTO (RIM). La SSII Wyzapps software a été fondée et est dirigée par Monsieur Franck MOUTARDIER, informaticien spécialisé dans la technologie IBM Lotus. Le société RIM a été crée par Monsieur Franck MOUTARDIER et Monsieur Sylvain GRENIER. Ma mission au sein de l'entreprise Route Inn' Moto (RIM) était de concevoir et maintenir un site web marchand.



La société RIM à été créée en mars 2008 et possède 4 domaines d'activité stratégique:

1. Distribution de motos 125 CM³ importées de chine sous la marque DRIM
2. Réparation de 2/4 roues (motos, quads, buggies, scooters)
3. Vente/reprise de motos, scooter, quads
4. Vente de produits dérivés : d'équipements (veste, gants, casques), pièces (motorisation) et accessoires (tuning)

Le premier domaine d'activité stratégique (DAS) est le cœur de métier de l'entreprise.

Figure 1 : Organigramme de la société



2/ Mes missions

Dans le cadre de mon année de formation j'ai été chargé de concevoir un site e-commerce, ainsi que le back office associé. De plus, j'étais en charge de la rédaction de l'ensemble des dossiers de la solution (cahier des charges, dossier de conception etc.). En outre, j'ai effectué toutes les étapes de conception et de mise en œuvre de la solution. Ainsi, j'ai décidé des méthodologies devant être appliquées, ainsi que du planning de développement. Par ailleurs, durant deux mois j'ai géré et planifié le travail d'un stagiaire développeur.

La liste des documents que j'ai élaborés est la suivante :

- Le cahier des charges : il contient les spécifications initiales des exigences du maître d'ouvrage.
Référence : RTN-CDC-V1
- Le modèle métier : il correspond au modèle conceptuel, et est accompagné des règles de gestions, et du modèle de communication.
Référence : RTN-MM-V1
- Le dossier de spécification : il contient les instructions techniques de mise en place de l'environnement et de développement de l'architecture.
Référence : RTN-DS-V1
- Le dossier des règles de gestion applicatives : il contient l'ensemble des règles de gestion de base mise en œuvre pour la conception de la solution.
Référence : RTN-RGA-V1
- Le dossier de conception : il contient l'ensemble des choix techniques et architecturaux de la solution.
Référence : RTN-DC-V1
- Le dossier de configuration : il contient l'ensemble des informations permettant de configurer la solution.
Référence : RTN-DCO-V1

3/ Analyse du projet

1. Règles de gestion et processus métier

En me basant sur le cahier des charges et les précisions des acteurs du projet j'ai établi une description du processus métier ainsi que la liste des principales règles de gestion.

Processus métier

Les étapes du processus métier pour la partie vente sont les suivantes:

- Définir le catalogue des articles en vente (vente, dépôt vente)
- Définir les catégories pour les articles
- Classer les articles par catégorie
- Rendre possible le classement par catégorie d'articles
- Publier le catalogue sur internet
- Fournir un caddie électronique pour l'établissement des commandes
- Fournir une liste des concessionnaires
- Recevoir et traiter les commandes des clients
- Vérifier la disponibilité des articles commandés
- Enlever les articles demandés de l'inventaire
- Envoyer la facture au client (par e-mail ou par courrier)
- Envoyer les commandes aux clients

Les étapes du processus métier pour la publication d'articles de presse sont les suivantes :

- Définir des types d'éditeur
- Définir des éditeurs
- Publier des articles de presse

Les étapes du processus métier pour la partie "zone client" sont les suivantes :

- Vérifier l'identité du client
- Fournir des formulaires pour la saisie et la modification des informations clients
- Fournir un historique des commandes
- Fournir la possibilité de télécharger les factures des anciennes commandes
- Fournir un suivi des commandes
- Fournir un moyen de contacter le service client

Règles de gestion

Les règles de gestion suivantes ont été dégagées :

- R1 : Une **catégorie d'articles** rassemble un ou plusieurs **articles du même type**. Un **article** appartient à une seule **catégorie**.
- R2 : Les articles de types **pièce** et **accessoire** peuvent être lié ou non à **un ou des véhicules** en base de données.
- R3 : Les **articles** de types **moto** doivent obligatoirement avoir des images de **carrousel**.
- R4 : Les **articles** peuvent être **vendu** en aucun ou plusieurs **couleurs**.
- R5 : Un **article** ne peut être **vendu** à un **prix** inférieur à 1€, il ne peut ne pas avoir de **TVA**.
- R6 : Deux **articles** de la **même catégorie** ne peuvent avoir la même **référence**.
- R7 : Un **article** ne peut être **vendu** que s'il en reste au moins un **exemplaire** en **stock**.
- R8 : Un **client** n'est rattaché qu'à un seul type de client.
- R9 : Deux **clients** ne peuvent avoir le même code de **référence**, le même **nom d'utilisateur**.
- R10 : Un **client** peut être enregistré sans posséder de **compte** validé pour l'**accès** depuis le site web
- R11 : Un **client** doit obligatoirement posséder une **adresse physique**.
- R12 : Un **client** ne peut être rattaché qu'à une seule **civilité**.
- R13 : Un **client** de type **professionnel** doit posséder un numéro de **TVA intracommunautaire**, une **raison sociale** et un **numéro de Siret**.
- R14 : Un **panier** est **obligatoirement** attaché à une **facture**.
- R15 : Une **facture** n'est rattachée qu'à un **type de facture**. Un type de **facture** n'est rattaché qu'à un seul **type de client**.
- R16 : Une **facture** possède un **statut de commande** (en attente, validée, refusée etc.).
- R17 : Une **facture** possède **éventuellement** un **transporteur**.
- R18 : Une **facture** possède un type de **moyen de paiement**.
- R19 : Un **type d'éditeur** de presse rassemble un ensemble d'**éditeur**.
- R20 : Un **éditeur** est rattaché à un **type d'éditeur**.
- R21 : Deux **éditeurs** ne peuvent avoir le même **nom** ou et la même **adresse web**.
- R22 : Un **article de presse** est rattaché à un **éditeur**. Deux articles ne peuvent avoir la même **adresse web**.
- R23 : Un **type de reprise** rassemble un ensemble de **reprise**.
- R24 : Un **type d'état de reprise** rassemble un ensemble de **reprise**.
- R25 : Une **reprise** est rattachée à un **type de reprise** et à un **type d'état de reprise**.
- R26 : Un **concessionnaire** doit avoir un **nom**.
- R27 : Deux **concessionnaires** peuvent avoir le même **nom**.

Par la suite j'ai utilisé ces règles pour effectuer une analyse MERISE de l'organisation, qui a ensuite découlé sur une analyse du système d'information et enfin sur la conception d'un MCD, et d'un MPD.

4/ Méthodologie

1. Analyse de l'organisation

Après avoir dégagé les principales règles de gestion ainsi que les processus métier, j'ai effectué une analyse de l'organisation selon la méthode MERISE. Pour m'aider dans les étapes de modélisation j'ai utilisé les logiciels Microsoft Visio, PowerDesigner et StarUML.

Identification des acteurs du système d'information

Tout d'abord j'ai procédé à une identification de l'intégralité des acteurs de l'organisation et donc du futur système d'information. Pour cela j'ai procédé par observation, écoute et interrogation des acteurs. De mêmes, en m'appuyant sur le cahier des charges les acteurs externes à l'organisation, mais interagissant avec elles ont pu être identifiés.

Il existe deux classes d'acteurs :

Les acteurs externes : Banque, Clients, Internaute.

Les acteurs internes : Service client, Service livraison et expédition, Service comptable, Service commercial, Service technique, Service informatique.

Certains des acteurs internes sont particuliers dans le sens où ils n'existent pas distinctement au sein de l'organisation, étant donné que celle-ci est restreinte ce qui fait que plusieurs personnes occupent plusieurs rôles. Néanmoins ces acteurs existent conceptuellement.

Figure 2 : Description des acteurs du SI

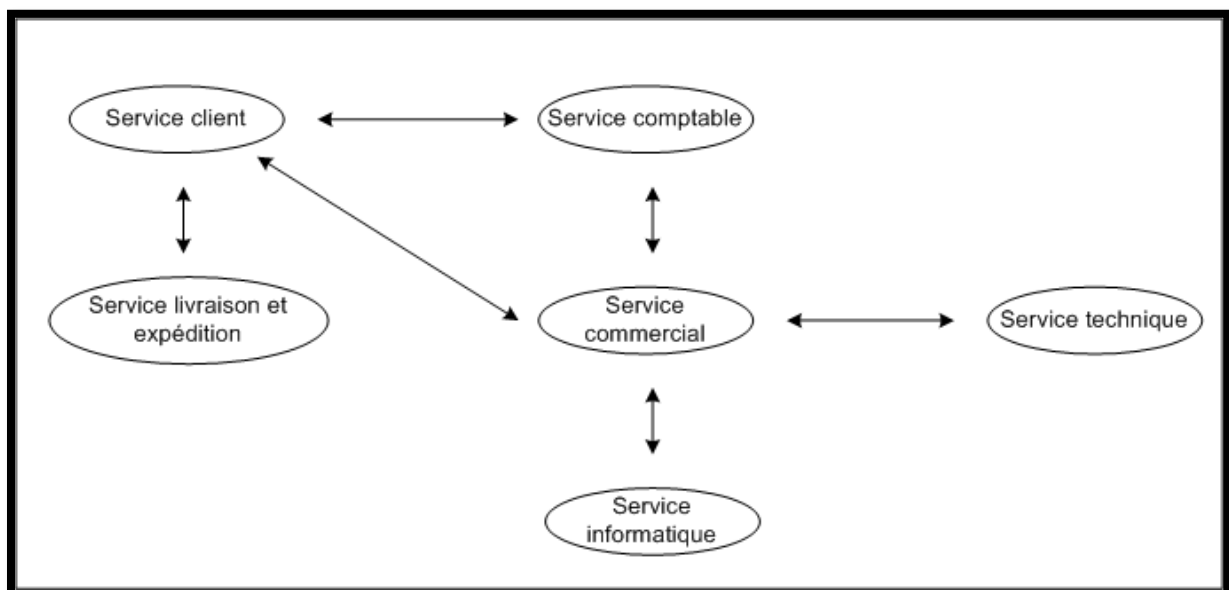
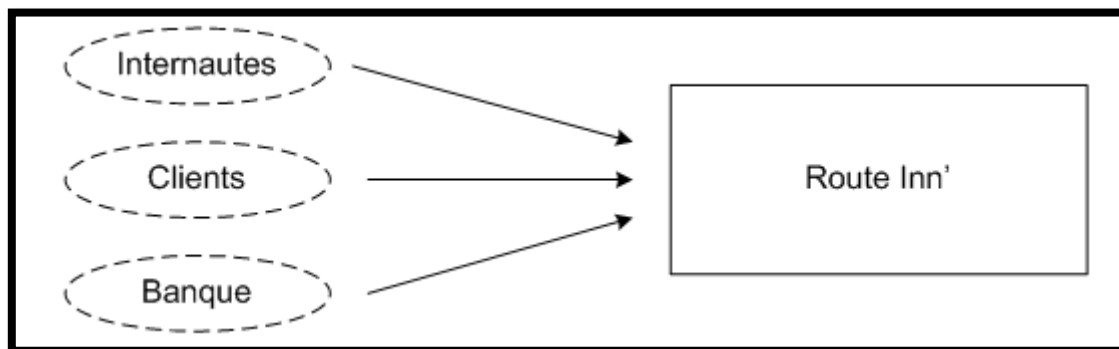
Nom	Définition
Service client	Ce service est en charge de la gestion des clients, des commandes et de la relation client.
Service livraison et expédition	Ce service gère la préparation, l'expédition et la livraison des commandes.
Service comptable	Ce service gère l'encaissement des paiements et tient la comptabilité.
Service commercial	Ce service est en charge de la promotion des produits.
Service technique	Ce service est en charge de l'entretien des véhicules, et de la réparation.
Service informatique	Ce service assure la gestion du parc informatique de l'entreprise et la conception des outils développés.
Clients	Les clients sont des internautes possédant un compte client.
Internautes	Les internautes sont les visiteurs du site marchand

J'ai poursuivi mon analyse en identifiant les flux d'information au sein de l'organisation, ainsi que les informations que celle-ci échange avec l'extérieur.

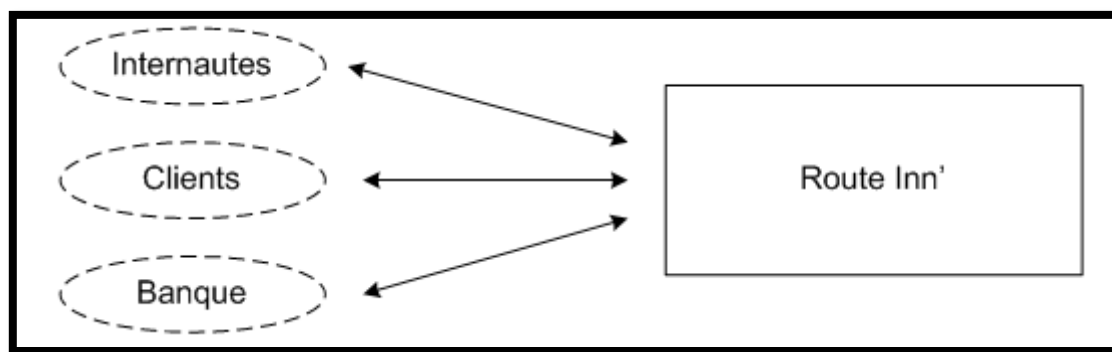
Communication au sein de l'organisation et avec l'extérieur

Pour identifier l'ensemble des flux j'ai commencé par définir l'organisation en utilisant des diagrammes de flux.

Définition de l'organisation



J'ai ensuite modélisé le contexte général dans lequel s'inscrit l'organisation à l'aide d'un diagramme de contexte auquel j'ai adjoint un tableau décrivant les flux entre les acteurs externes et l'organisation.

Diagramme de contexte

	Envoi	Retour
Internaute	Demande de renseignements	Renseignements de la part du service client
Clients	Demande de renseignements sur les produits Demande d'informations sur une ou des commandes	Renseignements de la part du service client
Banque	Retour d'informations concernant une commande	

Enfin j'ai construit le diagramme conceptuel de flux et l'ensemble des matrices de flux.

- Annexe 1 : Diagramme conceptuel de flux
- Annexe 2 : Matrice de flux entre les clients et les services de l'entreprise lors d'une commande
- Annexe 3 : Matrice de flux entre le client déposant un article en dépôt vente ou en reprise
- Annexe 4 : Matrice de flux entre les internautes et le service client
- Annexe 5 : Matrice de flux entre les services comptable, commercial et client

2. Utilisations de la solution par l'organisation et les acteurs externe

Parallèlement à ces travaux mes discussions avec le maître d'ouvrage, les acteurs de l'organisation et mes observations m'ont permis d'échafauder les diagrammes d'utilisations de la solution. Ces diagrammes prennent en compte l'ensemble des entités et des flux d'informations découlant de l'analyse de l'organisation.

L'ensemble des cas d'utilisation sont répertoriés dans le dossier de spécification de la solution RTN-DS-V1.

Figure 3 : Cas d'utilisation de l'authentification

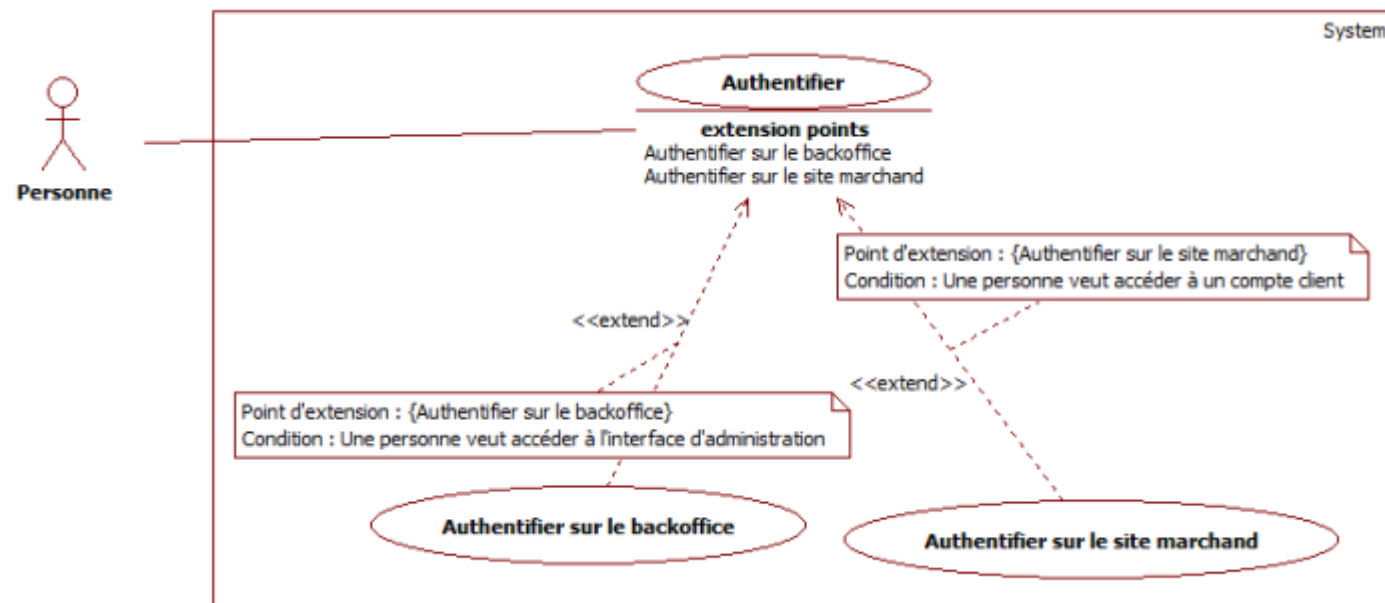
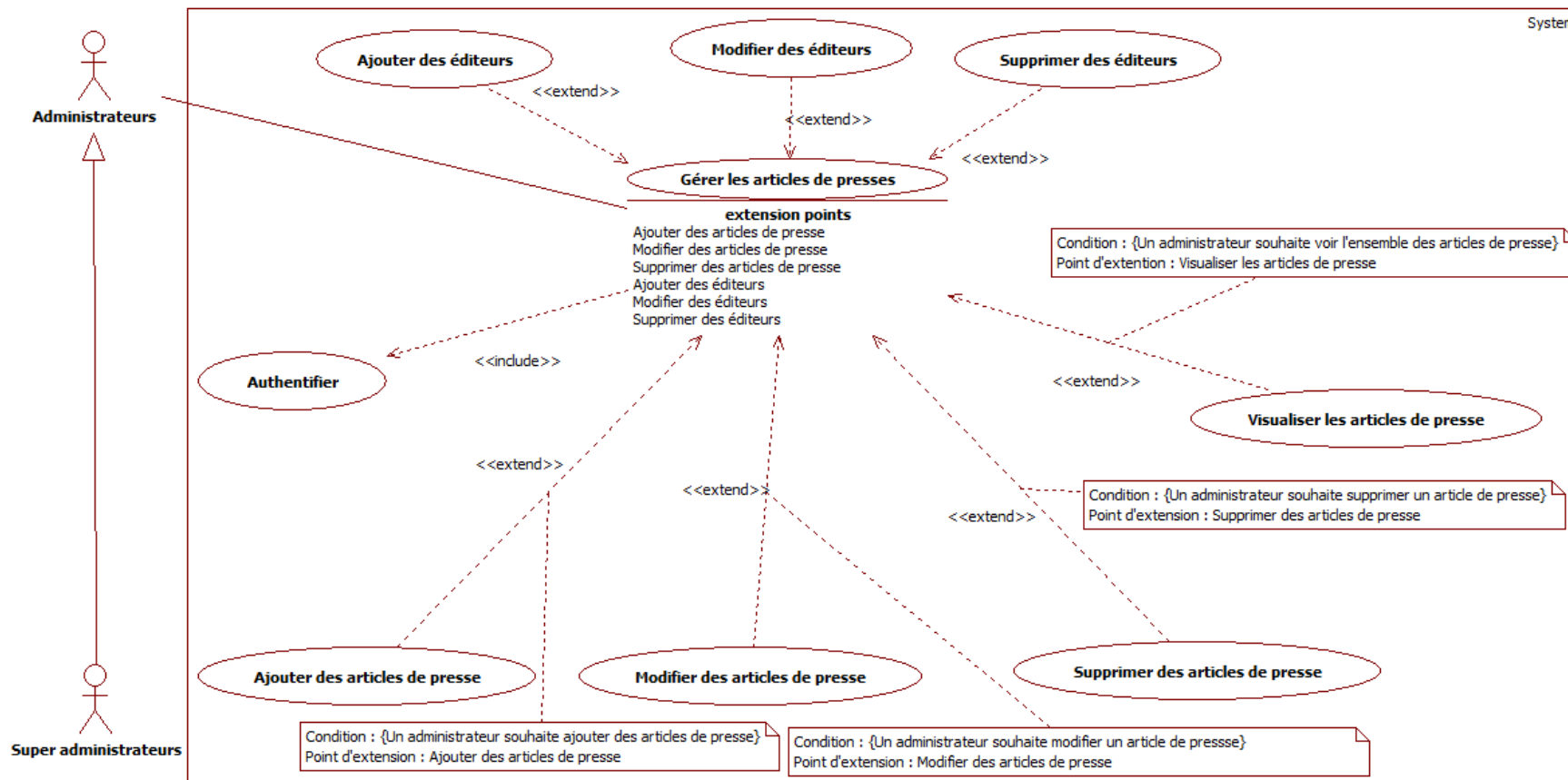


Figure 4 : Cas d'utilisation de la gestion d'article de presse



3. Conception de la base de données

Les éléments recueillis m'ont permis d'établir le modèle conceptuel de données. Pour établir, un modèle correcte je me suis aidé du livre (S. Lecompte, Mysql 5.0) expliquant la démarche MERISE appliquée à la modélisation de base de données, ainsi que l'utilisation du logiciel PowerAMC.

1. Identification des objets métiers

Pour établir le MCD j'ai commencé par dégager les principaux objets métiers (ou entités) qui représentent les composants acteurs du processus métier. Par la suite pour chacun des objets métiers j'ai identifié les attributs en m'aidant des règles de gestion et du processus métier ainsi que de questionnements auprès du maitre d'ouvrage. De plus j'ai convenu pour chaque objet d'un champs d'identifiant.

2. Identifications des relations

Dans un second temps j'ai identifié les relations (ou associations) existantes entre les entités. Pour cela, je me suis aidé des règles de gestion, du processus métier ainsi que questionnement auprès du maitre d'ouvrage. Afin de garantir que la base de données soit bien conçue, j'ai validé chaque étape de la conception grâce aux fonctionnalités de validation présentent dans le logiciel PowerDesigner que j'ai utilisé (Annexe 6).

Avant de passer à la conception du modèle physique de données, j'ai modélisé le Modèle Orienté Objet (Annexe 7) afin d'avoir une vue d'ensemble des relations entre les entités.

Pour concevoir le modèle physique de données j'ai utilisé la fonctionnalité de génération de script de base de données présente dans le logiciel PowerDesigner. J'ai ensuite procédé à une vérification du script généré, afin de m'assurer qu'il ne contienne pas d'erreurs.

3. Fiabilité et monté en charge

J'ai pris la décision d'utiliser des procédures stockées pour les requêtes sensibles telle que l'ajout d'une moto. En effet, l'utilisation de procédures stockées permet de faire ces opérations de façon transactionnelle, et donc de garantir la cohérence de la base de données.

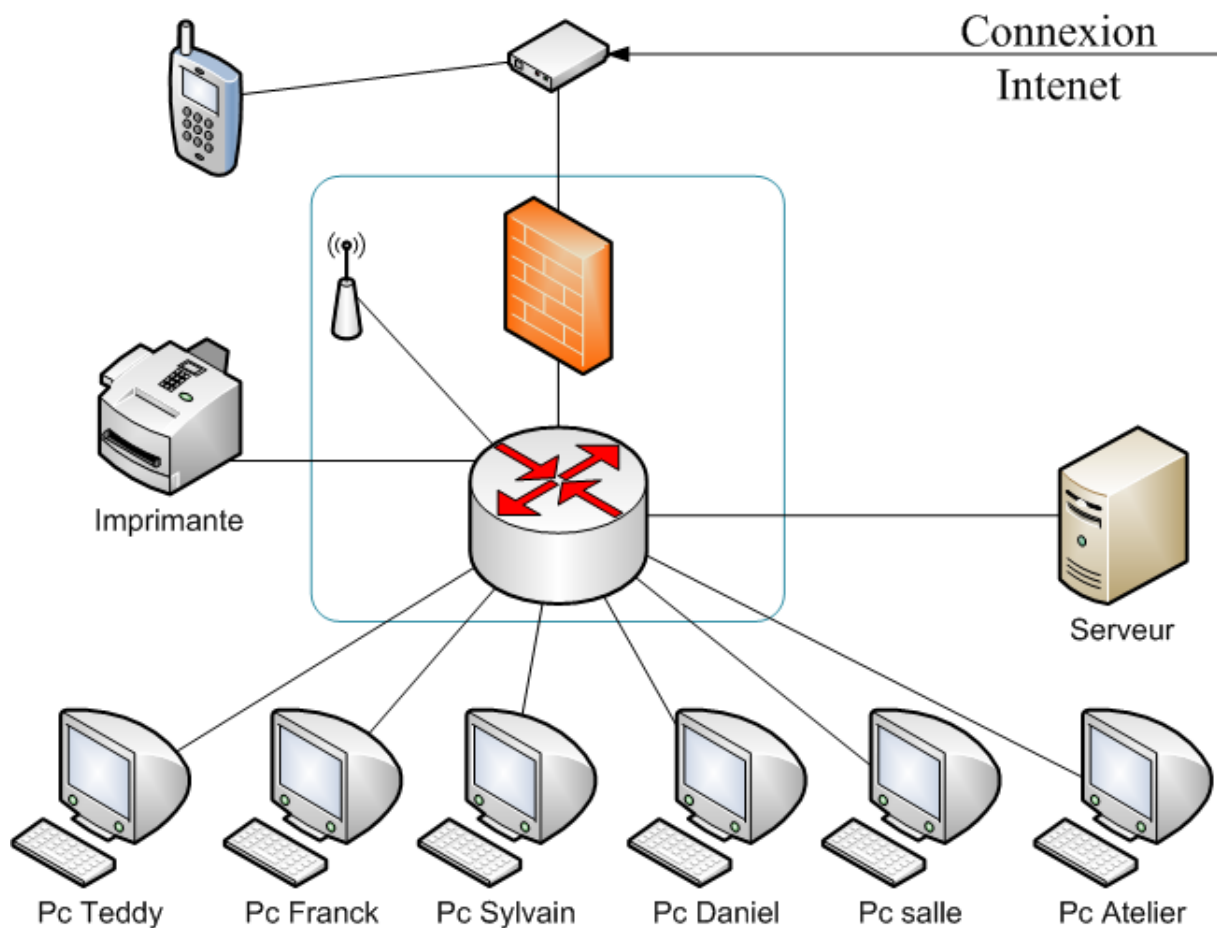
En outre, j'ai privilégié l'utilisation de vues pour certaines requêtes comportant un nombre important de jointures. L'utilisation de vues permet au serveur SQL de ne pas avoir à recalculer le plan d'exécution de la requête à chaque fois. De ce fait la charge du serveur diminue.

5/ Réalisation du projet

1. Technologies et outils

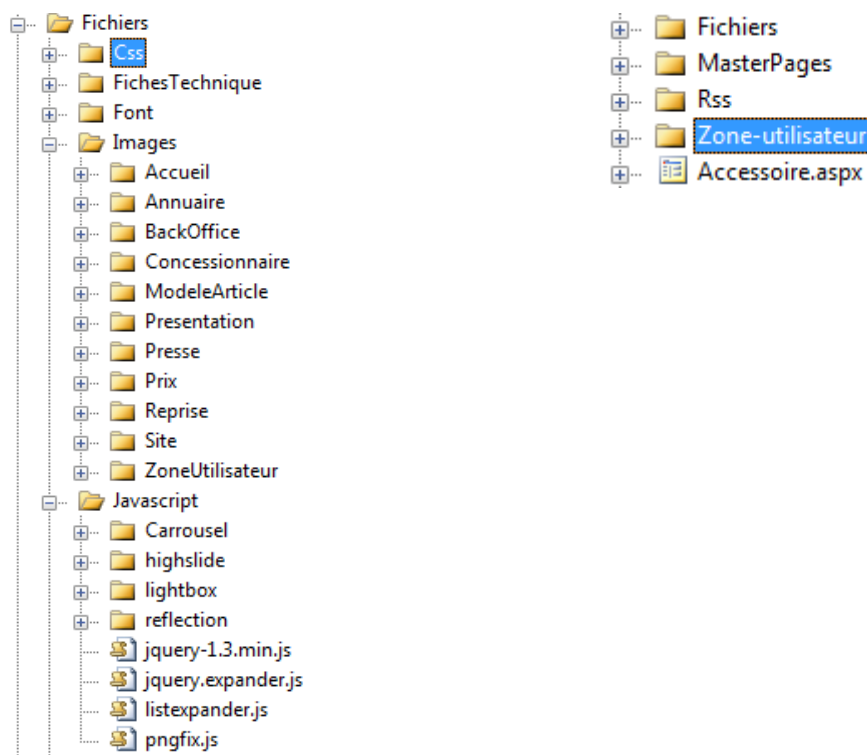
Pour réaliser ce projet j'ai utilisé le langage C# 3.5, ainsi que la plateforme ASP.net 3.5. De plus j'ai travaillé avec les bases de données SQL server 2005 et 2008, le développement s'est effectué sous Visual studio 2008. A la demande du maître d'ouvrage qui souhaitait utiliser un outil tel que Hibernate j'ai choisi d'utiliser Linq To Sql, en me fondant notamment sur ma compétence et l'état des outils disponibles sur la plateforme notamment Entity Framework.

Figure 5 : Topologie de l'environnement de l'entreprise

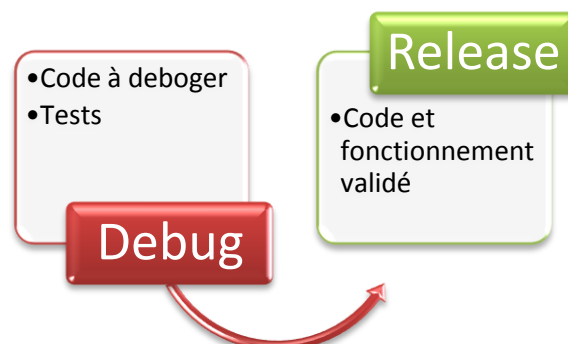


2. Organisation de la solution

Avant de commencer la programmation de la solution j'ai déterminé l'arborescence des dossiers. J'ai choisi de créer un dossier Administration pour l'ensemble des fichiers de l'interface d'administration, les fichiers du site marchand se trouvant à la racine du dossier principal. L'ensemble des ressources externes aux pages web se trouvent dans le dossier « Fichiers » contenant lui même un dossier par type de ressource. Par ailleurs, un dossier « Zone-utilisateur » contient l'ensemble des pages ayant trait à la gestion de compte client, et à la passation de commandes. De même un dossier Rss contient les handlers ASP.net fournissant les flux RSS. Enfin un dossier MasterPages contient l'ensemble des masterpages de la solution.



La solution de développement a été scindée en deux branches l'une servant au débogage et aux tests, et une autre étant la branche principale contenant le code finale de la solution.



3. Intégration graphique

Le designer a réalisé les maquettes du site marchand avant le début du projet, à l'aide du logiciel Adobe Photoshop. J'ai effectué le travail d'intégration graphique en collaboration avec un stagiaire développeur, nous avons réalisé ce travail en respectant les décisions prises lors de la réaction du dossier de conception (RTN-DC-V1).

Pour nous assurer de la qualité du travail effectué, nous avons validé chacune des maquettes en suivant le cycle suivant.

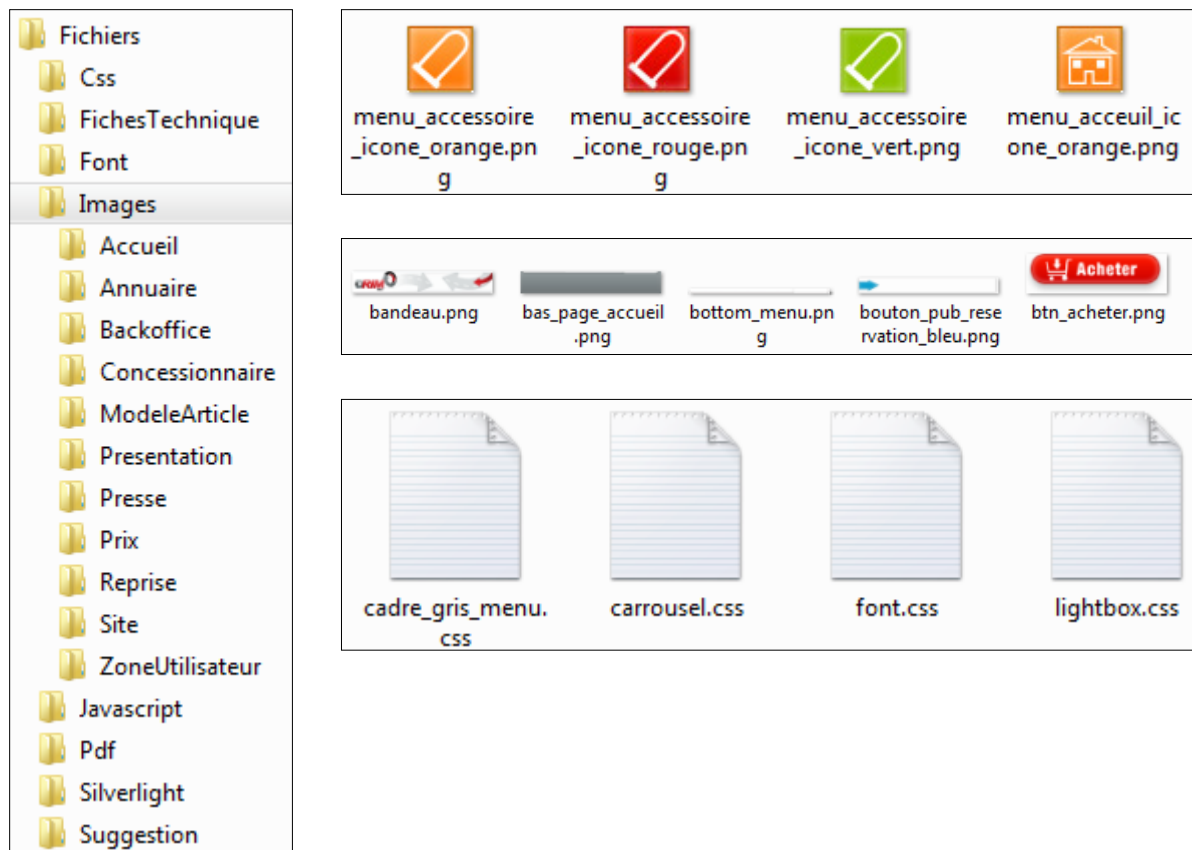


La validation des différentes étapes était effectuée par nous même développeurs, mais aussi par le maître d'ouvrage et par le designer. Cette méthodologie nous a permis d'assurer la meilleure qualité possible. En outre, nous avons mis en place une architecture de dossier simple et logique, veillé au respect des conventions de nommage ainsi qu'au format de fichiers utilisés.

L'utilisation du format PNG pour l'ensemble des images composants le site, pose des problèmes sur les versions inférieures à internet explorer 7 avons mis en place un script JavaScript résolvant le problème de non gestion de la transparence.

```
<!--[if lt IE 7]>  
<script defer type="text/javascript" src="Fichiers/Javascript/pngfix.js">  
</script><![endif]>
```

Figure 6 : Arborescence et conventions de nommage et formats de fichiers



Durant la conception des maquettes du site marchand nous avons rencontré des problèmes liés aux polices utilisés sur le site, à savoir Helvetica Neue. En effet, il s'est avéré que seul le navigateur Internet Explorer supportait les fichiers de polices, car c'est ce dernier qui est à l'origine du balisage CSS :

```
@font-face
{
  font-family:"HelveticaNeueLT Std Blk Cn";
  src: url('../Font/HelveticaNeueLTStd-BlkCn.otf');
}
```

De plus l'utilisation de police au format "naturel" ne vas pas sans poser certains problèmes d'ordre juridique du à la licence des polices Helvetica Neue qui interdisent leur diffusion. Là encore après enquête nous avons découvert que seul l'éditeur de Internet Explorer avait pris en compte le problème juridique en mettant en place un format spécifique permettant le respect des licences des éditeurs de polices tout en permettant l'utilisation de polices telles que Helvetica Neue. Ce format nommé Embedded Open Type a été soumis au W3C mais n'a pas encore été normalisé (cf : <http://www.w3.org/Submission/EOT/>).

Nous avons donc contourné cet obstacle, en transformant le texte utilisant ces polices en image. L'utilisation d'images au lieu de texte posant des problèmes de référencement c'est pourquoi l'attribut ALT de chaque image reprend l'intégralité du texte qu'elle présente.

Après avoir assemblé les maquettes XHTML, nous nous sommes concertés avec le maître d'ouvrage pour décider des balises de méta données que nous devions ajouter aux pages maîtres.

```
<meta name="reply-to" content="contact@routeinn.com" />
<meta name="Revisit-after" content="4 days" />
<meta name="author" content="RouteInn'" />
<meta name="robots" content="all" />
<meta name="rating" content="general" />
<link rel="start" title="Accueil" href="Default.aspx" />
<link rel="shortcut icon" type="image/x-icon" href="/icone.ico" />
<link rel="icon" type="image/png" href="/icone.png" />
<meta name="verify-v1"
content="8Bnh6ojeKzET4FHfFHF3O5qsI1lk/yVqHCnfz3wGNJg=" />
<meta name="msvalidate.01" content="1FA23BB243FF1F21C92D4C6E314C90DF"
/>
<link rel="canonical" href="http://www.drim-moto.com" />

    <meta name="DC.Description" content="Route Inn' Moto, créateur de
solution en milieu urbain. Découvrez les 125 DRIM, Roadster et Sport.
Ventes, reprises, réparations, distribution motos, pièces, accessoires et
équipements sur toute l'Ile de France." />
    <meta name="DC.Keywords" content="drim, dream, drime, moto, motos,
125cc, moto 125, routeinn, route inn, moto neuve, motos neuves, achat
moto, moto chinoise, motos chinoises, drym, vend moto, 125cm3, motos
125cm3, magasin moto paris" />
    <script type="text/javascript">
        var gaJsHost = (("https:" == document.location.protocol) ?
"https://ssl." : "http://www.");
        document.write(unescape("%3Cscript src='" + gaJsHost + "google-
analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));
    </script>

    <script type="text/javascript">
        try {
            var pageTracker = _gat._getTracker("UA-7462xxx-1");
            pageTracker._trackPageview();
        } catch (err) { }
    </script>
```

Le designer n'ayant pas créé de maquette pour le Back office j'ai pris la conception de l'interface de celui-ci à ma charge. J'ai décidé d'adopter une présentation classique avec une barre de menu principaux en haut de page, une seconde horizontale barre de contenant les sous-menus éventuels, une large zone verticale à gauche qui comporte l'ensemble des commandes et la présentation des données, et enfin une barre verticale à droite comportant les actions spécifiques à la section. Le "look & feel" du backoffice a été emprunté au logiciel windows media player 12.

Figure 7 : Agencement du back office



Administration Clients Commandes Articles Concessionnaires Reprises Presse									
Tri par code postal		95870	Voir l'ensemble des clients		Recherche				
Reference	Nom	Prenom	Telephone	Ville	Email	Suppr.	Modif.	Voir	
1420091201	Albina	Teddy	0173652375	Bezons	teddyalbina@hotmail.fr	Suppr.	Modif.	Voir	Voir la liste des clients
1Pa20091201	Palisse	Cindy		Bezons		Suppr.	Modif.	Voir	Ajouter des clients

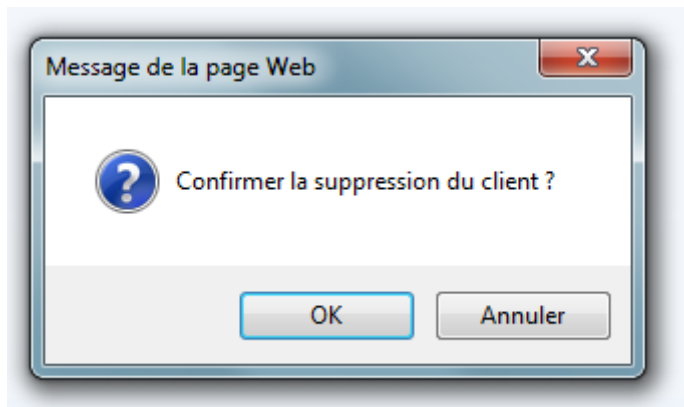
Chaque section du backoffice, possède une page listant l'ensemble des données manipulées par celle-ci. Ces données sont affichées à l'intérieur d'un gridview, dont la mise en forme est toujours la même afin de garantir la cohérence de l'interface, et une meilleure compréhension de cette dernière par l'utilisateur.

Chacun de ces gridview comporte les options « voir », « supprimer » et « modifier ». La fonctionnalité voir permet de visualiser l'ensemble des informations sur le type d'information traité dans la section. L'option « supprimer » permet d'ôter de la base de données, l'enregistrement sélectionné. L'option « modifier » permet quant à elle de modifier les données de l'enregistrement sélectionné. Par ailleurs, l'option supprimer est soumise à une validation, afin que l'utilisateur ne supprime pas involontairement des informations de la base de données.

Figure 8 : Gridview de listing des clients

Reference	Nom	Prenom	Téléphone	Ville	Email			
1AI20100101	Albina	Teddy		Bezons		Suppr.	Modif.	Voir

Figure 9 : Fenêtre de confirmation de la suppression d'un client



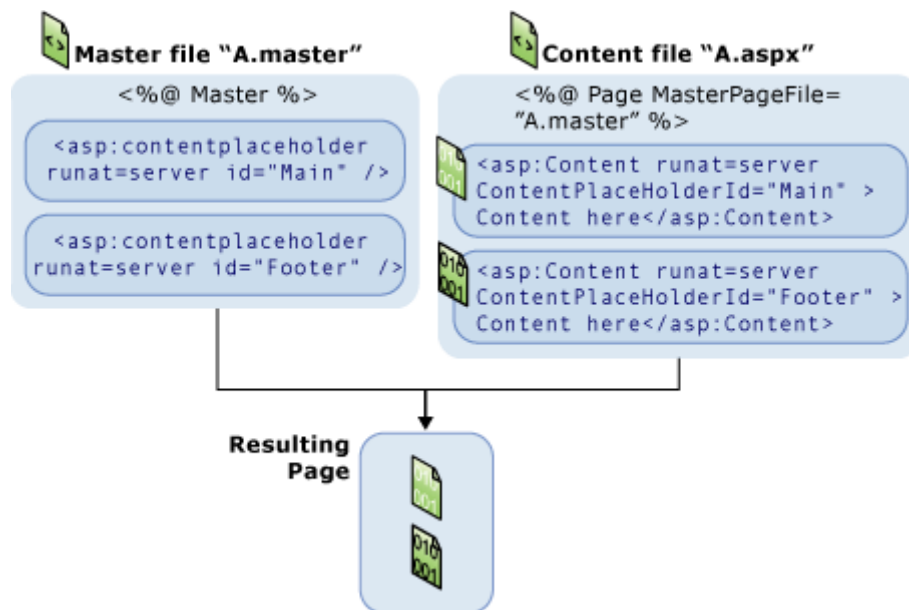
1. Mise en place des templates

L'ensemble des éléments issus de l'intégration graphique précédente ont ensuite été utilisés pour mettre en place les templates du site marchand. Pour cela nous avons respecté l'architecture ASP.net en nous appuyant sur des masterpages. Les masterpages permettent de ne pas avoir à refaire le travail d'intégration pour chaque page du site et assurent donc sa cohérence graphique.

Le système de masterpage fonctionne avec des zones dites zone de contenu ; les masterpages étant quant à eux appelés page de contenu (cf : [http://msdn.microsoft.com/fr-fr/library/wtxbf3hh\(VS.80\).aspx](http://msdn.microsoft.com/fr-fr/library/wtxbf3hh(VS.80).aspx)). Ces zones sont délimitées par des balises ContentPlaceholder.

```
<asp:ContentPlaceholder ID="ContainerCentral" runat="server" />
```

Figure 10 : Fonctionnement des masterpages



Pour obtenir des masterpages strictement identiques à l'intégration statique, nous avons effectué de nouveaux zoning afin de repérer les zones de contenus.



Une fois ce repérage effectué nous avons déplacé le code des pages statiques vers les masterpages. Après concertation nous avons décidé de mettre en place trois masterpages, un masterpage pour la page d'accueil, un masterpage pour le reste du site à l'exception des parties du site se trouvant dans la zone-utilisateur, et un masterpage pour la partie du site dédié aux clients (commandes, gestion de compte).

De plus nous avons déterminé l'existence de 6 zones de contenus sur le masterpage de la page d'accueil. Sur le masterpage dédié au reste du site nous avons déterminé l'existence de 4 zones de contenus. Enfin, sur le masterpage de la zone-utilisateur nous avons repérés 3 zones de contenus. Nous avons procédé de la même manière pour le backoffice, pour lequel il existe, 3 zones de contenus.

2. Utilisation des templates

Une fois les templates créés, il suffit pour chacune des pages de charger la ou les zones que l'on souhaite éditer. Pour cela la page (.aspx) doit dériver du masterpage souhaité.

```
<%@ Page Title="Route Inn' - Les échos de DRIM dans la presse"
Language="C#" MasterPageFile="~/MasterPages/Site.master"
AutoEventWireup="true" CodeFile="Presse.aspx.cs" Inherits="Presse" %>
```

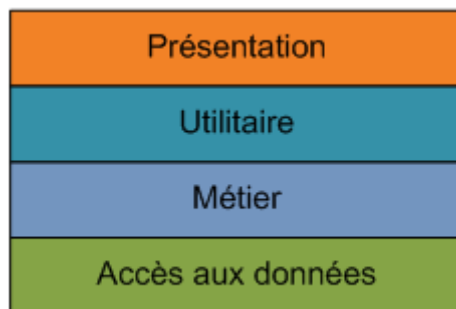
L'édition d'une zone de contenu se fait alors de la façon suivante.

```
<asp:Content ID="Content3" ContentPlaceHolderID="ContainerCentral"
runat="Server"> contenu </asp:Content>
```

4. Architecture

J'ai choisi d'orienter la solution vers une architecture à quatre couches, afin d'assurer son évolutivité ainsi qu'une maintenance plus aisée. L'utilisation de Linq To Sql est un peu particulière dans le sens où LINQ est la couche d'accès aux données. En effet, cette technologie fait qu'il est possible d'adresser directement des requêtes à la base de données en utilisant directement les classes générées, car Linq est un langage d'interrogation de données à part entière.

Figure 11 : Architecture de la solution



Couche d'accès aux données : cette couche contient l'implémentation des accès à la base de données afin de la masquer à la couche métier. Cette couche est en partie gérée par Linq To Sql.

Couche métier : cette couche contient les objets métier de l'application. Ces objets implémentent les fonctionnalités, et font le lien entre l'interface utilisateur et la couche d'accès aux données.

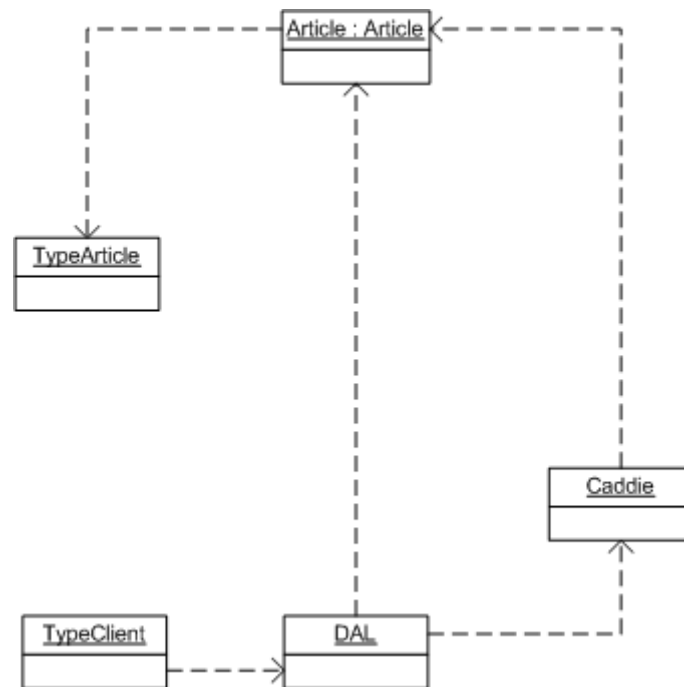
Couche utilitaire : cette couche contient les objets de validation, de génération des données.

Couche présentation : cette couche représente les interfaces des applications. Elle se charge d'afficher des contenus à l'utilisateur et de lui offrir des interfaces pour interagir avec l'application. La couche présentation repose sur l'architecture **Web Form** qui permet de séparer le code de présentation du code fonctionnel.

1. Composants

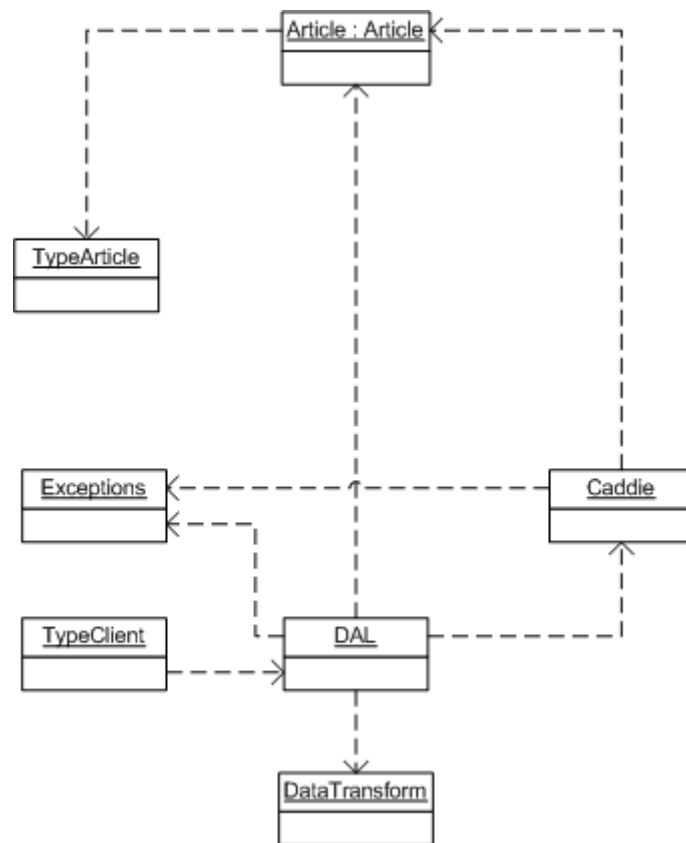
Une fois le choix de l'architecture arrêté, j'ai effectué une analyse UML de la solution afin de dégager l'ensemble des composants devant assurer son fonctionnement. Pour définir les composants j'ai procédé à un découpage des grandes fonctionnalités et entités du système (caddie, article, type d'article, accès aux données) ; Me permettant ainsi de dégager les principaux composants du système.

Figure 12 : Diagramme de domaine primaire



J'ai ensuite affiné mon analyse en m'interrogeant sur les fonctionnalités redondantes (contrôle de format de données etc.), c'est ainsi que j'ai défini le composant *RouteInn.Common* qui regroupe entre autre les énumérations de type d'article et de type de client ainsi que le composant *RouteInn.DAL.DataTransform*. Les composants ASP.net spécialisé ont été "découvert" au cours de la programmation.

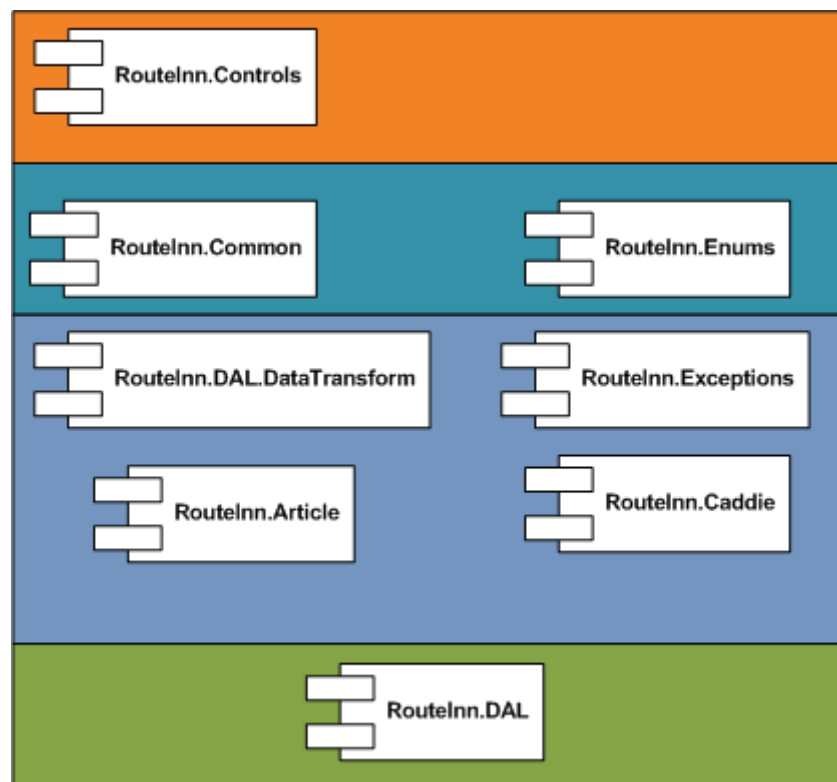
Figure 13 : Diagramme de domaine secondaire



Afin de m'assurer de la pertinence de mon analyse je me suis aidé d'outils pédagogiques notamment d'un livre sur UML (B. Charroux, A. Osmani, Y T-Mieg, 2008) et de ressources en ligne.

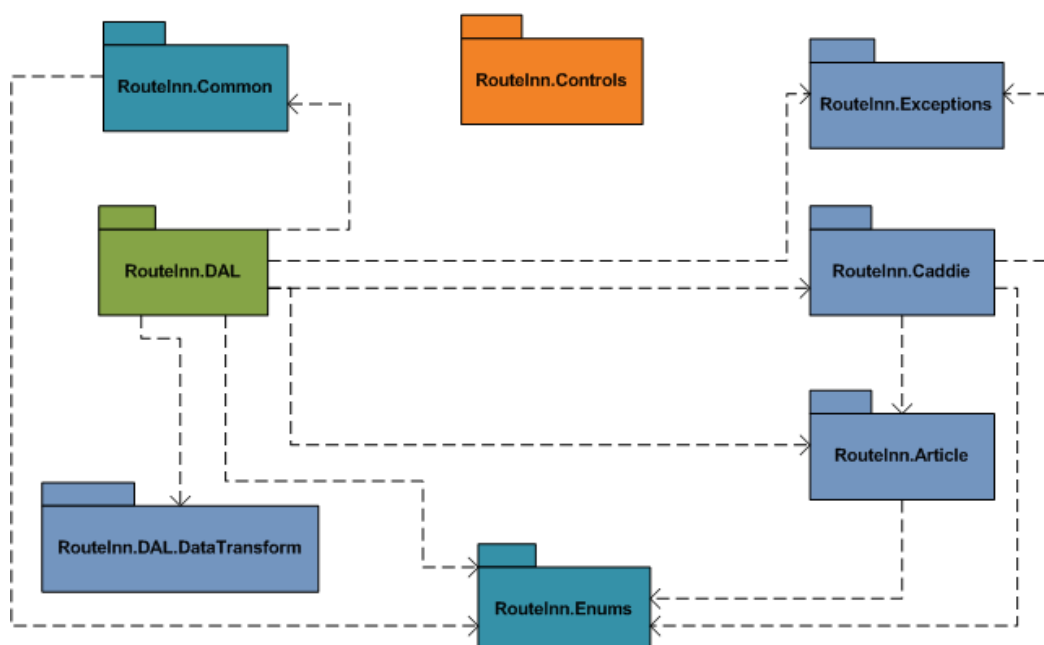
Après cette phase d'analyse, j'ai placé chaque objet dans un package distinct, afin d'assurer la cohérence et la compréhension de la solution. En outre, chaque composant respecte la convention suivante : *RouteInn.NomDuComposant*. Enfin, j'ai défini pour chaque composant la couche logicielle à laquelle il appartient.

Figure 14 : Emplacement des composants dans chacune des couches



En outre, j'ai créé un diagramme de dépendance des composants afin de m'assurer qu'il n'existe qu'un moindre nombre de dépendances circulaires, pouvant à terme entraver l'évolutivité de la solution.

Diagramme 1: Diagramme de dépendance des composants



2. Mise en œuvre

Avant de commencer la programmation des composants j'ai en premier lieu, dessiné le diagramme de classe de chacun d'eux (Annexe 8 et 9). Pour cela j'ai utilisé les logiciels, StarUML et PowerDesigner. La conception des composants s'est faite en utilisant les diagrammes de cas d'utilisation, afin de savoir quels services chacun d'eux devait rendre.

Tout d'abord j'ai modélisé les classes représentant les objets de base (Figure 10 et 11). Puis j'ai créé les énumérations des types d'articles et des types de clients. De même, j'ai ensuite modélisé les classes de gestion des exceptions. Le composant « RouteInn.Common » comportant des méthodes utilisant des expressions régulières, j'ai développé celles-ci à l'aide du langage PERL. La modélisation de la classe fournissant les méthodes d'accès aux données s'est faite au fur et à mesure de la programmation. En effet, j'ai programmé ces méthodes en fonction des besoins des classes supérieures, je n'avais au début du projet aucune réelle visibilité sur les méthodes nécessaires.

En outre, il s'est avéré que certaines fonctionnalités ne pouvaient être présentées à l'intérieur de pages classique (ex : gestion des couleurs). J'ai donc déporté ces dernières à l'intérieur de « user control asp.net ». Puis j'ai créé pour chacun d'eux une page asp.net les hébergeant. Enfin, l'utilisation du framework highslide m'a permis de présenter ces fonctionnalités sous forme de fenêtre flottante.

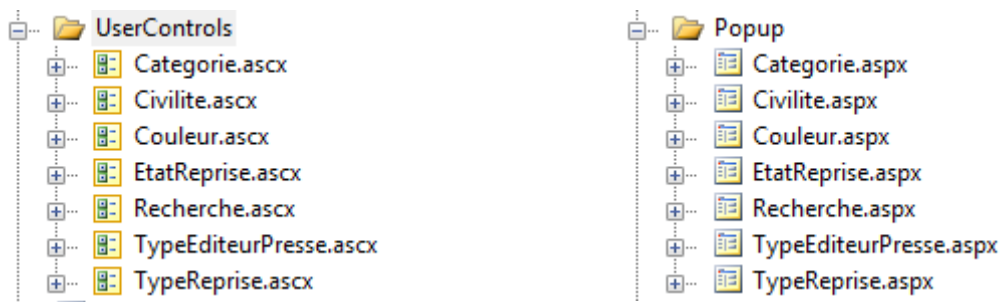


Tableau 1: Chargement d'un control utilisateur

```
<%@ Register Src="../UserControls/Categorie.ascx" TagName="Categorie"
TagPrefix="uccategorie" %>
```

Tableau 2 : Utilisation d'un control utilisateur

```
<uccategorie:Categorie ID="Categorie1" runat="server" />
```

Au cours de la mise en œuvre j'ai pris en compte la problématique du référencement, pour cela j'ai utilisé les Google webmasters tools, Google Analytics, ainsi que les outils du moteur Bing. Ces outils permettent aux moteurs de recherche d'indexer au mieux le site. De même, j'ai utilisé un fichier sitemap, ainsi qu'un fichier de contrôle robots.txt. Par ailleurs, j'ai mis en place les balises méta fournies par ces outils et d'autres dont j'ai décidé.

Nom	Valeur
verify-v1	8Bnh6ojeKzET4FHfFHF3O5qsl1lk/yVqHCnfz3wGNJg=
canonical	http://www.drim-moto.com
msvalidate.01	1FA23BB243FF1F21C92D4C6E314C90DF
start	<link rel="start" title="Accueil" href="Default.aspx" />
rating	<meta name="rating" content="general" />
reply-to	<meta name="reply-to" content="contact@routeinn.com" />
Revisit-after	<meta name="Revisit-after" content="4 days" />
author	<meta name="author" content="RouteInn" />
robots	<meta name="robots" content="all" />

J'ai à plusieurs reprises au cours de la mise en œuvre, mis à jour les diagrammes de classes et la documentation. En effet, certaines contraintes ou exigences ont été découvertes au cours de du développement.

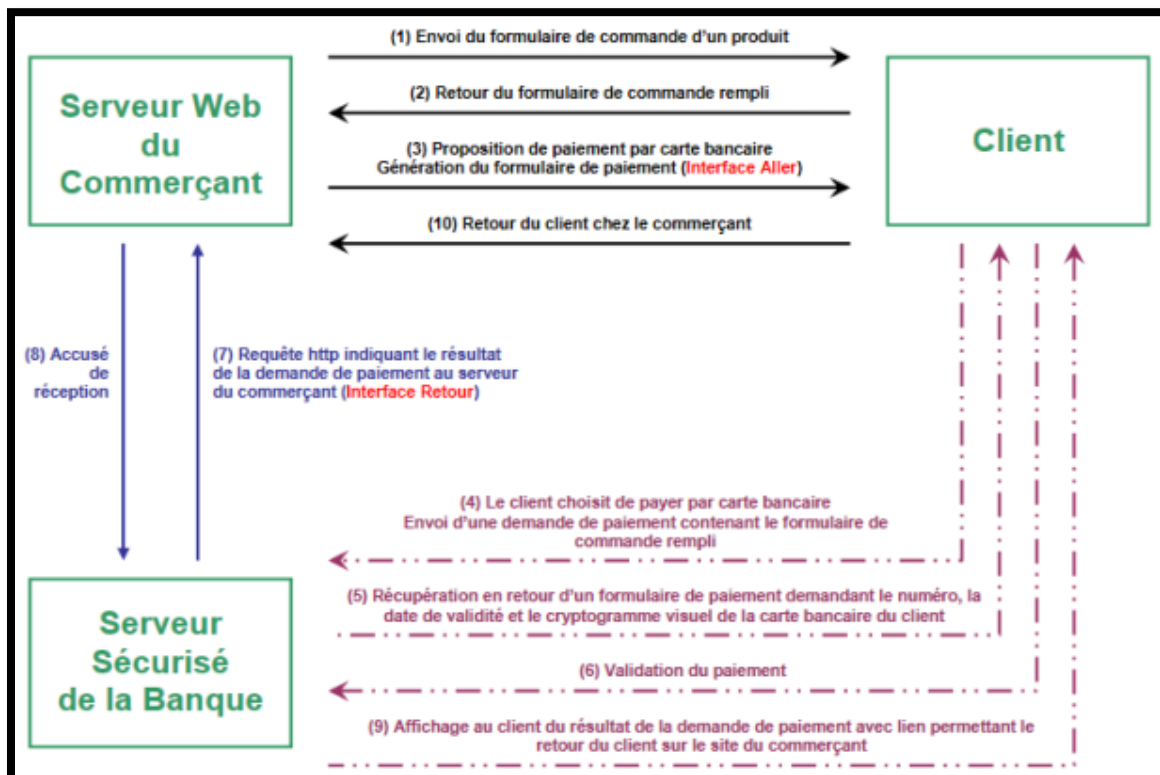
L'une des difficultés de la mise en œuvre de la solution a été de gérer les imprécisions de certaines demandes ou la survenue de nouvelles demandes. Par ailleurs, il m'a fallu gérer les demandes de changements de certaines fonctionnalités en cours de développement par le maître d'ouvrage. Ces contraintes ont engendrées des retards, en me contraignant à revenir sur les analyses que j'avais effectuées auparavant et/ou en m'obligeant à recommencer le développement. En outre, certaines fois je devais remettre à jour l'ensemble de la documentation.

5. Intégration du système de vente en ligne

Le site marchand permet aux clients de passer des commandes, le service de vente en ligne est fourni par la banque. Néanmoins le service bancaire, bien qu'externe à la solution interagît avec elle par le biais de page de retour d'informations spécialement mises en place en son sein.

La banque nous a fourni les documents de spécifications de sa solution de vente en ligne.

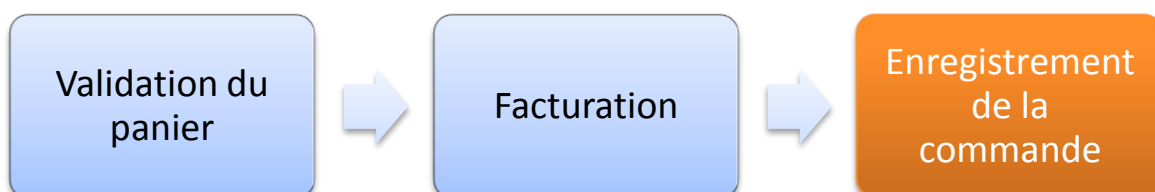
Figure 15 : Cinématique du paiement en ligne (fournit par la banque)



Informations complémentaires :

- RTN-BQG-V1.2
- RTN-BQT-V1.2

Tout d'abord j'ai schématisé les grandes étapes du processus de vente



Puis j'ai déterminé les informations importantes devant apparaître sur la page de facturation.

- Information du client (nom, prénom, adresse de facturation, numéro de téléphone, mail ...)
- Liste des transporteurs
- Numéro de facturation

De même j'ai défini avec le maître d'ouvrage les informations que le client doit pouvoir modifier lors de la commande. En outre, le maître d'ouvrage m'a communiqué le format que doit respecter le numéro de facturation, ainsi que les informations sur les données devant être conservées en base de données.

Figure 16 : Informations complémentaires concernant les données de la commande

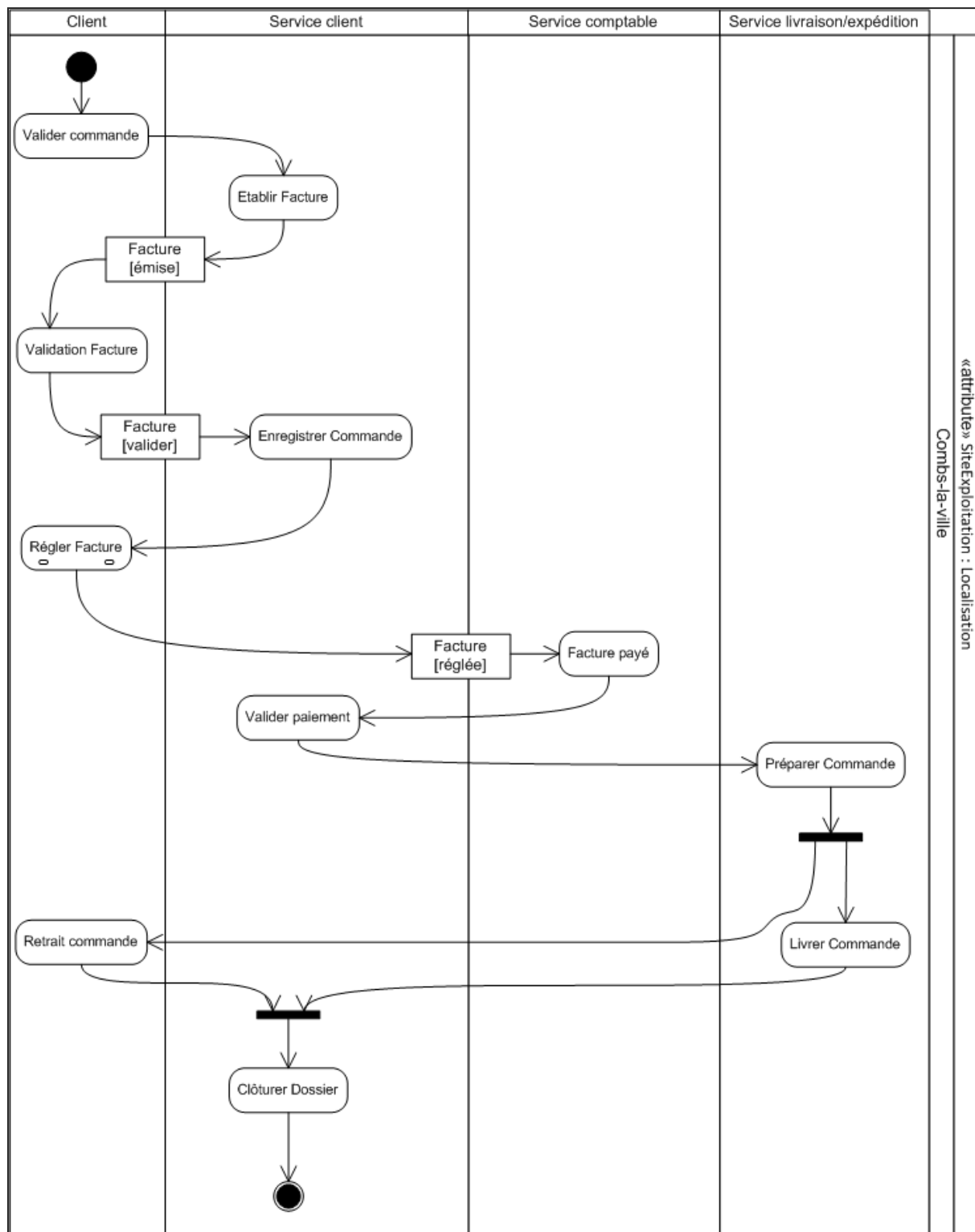
Information	Commentaire
Identifiant du moyen de paiement	Le TPE ne permet le paiement que par carte, mais un client peut payer en chèque par exemple la saisie des commandes est alors faite manuellement via le backoffice
Identifiant du statut de la commande	Le statut commande représente l'état de la commande à un instant T. Il peut être sur confirmé, en attente etc.
Identifiant du transporteur	Si le client choisit de se faire livrer il est impératif de conserver cet indicateur
Référence de la facture	Au format AAAAMMNNNN. AAAA = quatre chiffres de l'année MM = deux chiffres du mois NNNN = NNNN représente un chiffre quelconque unique.
Commentaire	Commentaire laisser sur cette vente via le backoffice mettre a NULL
Pour chaque commande le client pour spécifier une adresse de facturation et de livraison différente de celle enregistrée sur son compte client.	
Adresse de facturation	Adresse de facturation pour cette commande
Adresse de livraison	Adresse de livraison pour cette commande
Code postal	Code postal pour cette livraison
Ville	Ville pour cette livraison

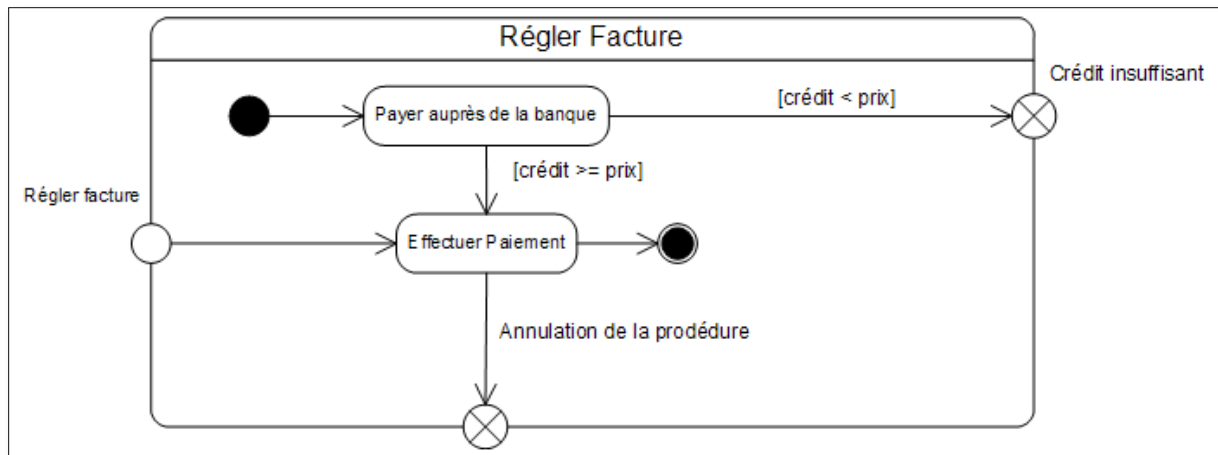
Informations complémentaires :

- RTN-MM-V1
- RTN-PD-SQL

Par ailleurs, j'ai utilisé le logiciel Visio pour modéliser les différentes étapes de la commande au sein de l'organisation et au sein de la banque grâce à des diagrammes d'activités.

Figure 17 : Diagrammes d'activités de la commande





Cette phase d'analyse m'a permis d'avoir une vue plus complète des différentes étapes de la vente en ligne, et des interactions du système d'information de la banque, avec le site marchand.

Les exemples de code et les spécifications fournies par la banque montrent que certains caractères de la table ASCII doivent être transposés en leur représentation html. Pour cela j'ai créé un dictionnaire de chacun de ces caractères et de sa représentation html.

```

    /// <summary>
    /// Dictionnaire des caractères ASCII à transposer
    /// </summary>
    private static readonly Dictionary<string, string>
    s_AsciiTransposeTable = new Dictionary<string, string>()
    {
        { ":", "&#x3a;" },
        { "/", "&#x2f;" },
        { "?", "&#x3f;" },
        { "=", "&#x3d;" },
        { "!", "&#x21;" },
        { "\", "&#x22;" },
        { "#", "&#x23;" },
        { "$", "&#x24;" },
        { "%", "&#x25;" },
        { "&", "&amp;" },
        { "'", "&#x27;" },
        { "(", "&#x28;" },
        { ")", "&#x29;" },
        { "*", "&#x2a;" },
        { "+", "&#x2b;" },
        { ",", "&#x2c;" },
        { ";", "&#x3b;" },
        { "<", "&#x3c;" },
        { ">", "&#x3e;" },
        { "[", "&#x5b;" },
        { "\", "&#x5c;" },
        { "]", "&#x5d;" },
        { "^", "&#x5e;" },
    }
  
```

```

        { "~", "&#x7e;" },
        { "{", "&#x7b;" },
        { "|", "&#x7c;" },
        { "}", "&#x7d;" }
    };

```

Puis j'ai écrit une fonction chargée de transposer les caractères ASCII en leur représentation html en utilisant le dictionnaire.

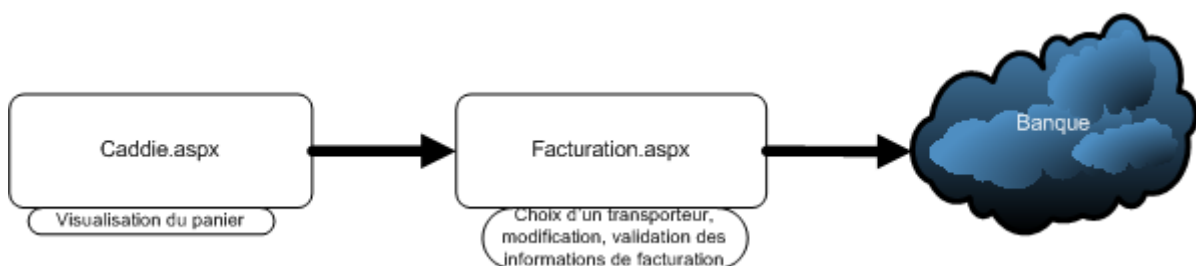
```

/// <summary>
/// Transcode les caractères ASCII en leur représentation Html
/// hexadecimal
/// </summary>
/// <param name="value">Chaine de caractère à transcoder</param>
/// <returns>Chaine de caractère transcodée</returns>
public static string TransposerAsciiCIC(string value)
{
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < value.Length; i++)
    {
        if
(s_AsciiTransposeTable.ContainsKey(value[i].ToString()))
            sb.Append(s_AsciiTransposeTable[value[i].ToString()]);
        else
            sb.Append(value[i].ToString());
    }

    return sb.ToString();
}

```

L'ensemble du processus de vente a été divisé en deux pages web, l'une présentant au client le contenu de son panier, et l'autre lui permettant de sélectionner un transporteur, et de modifier ses informations de facturation et/ou de livraison.



6. Intégration de composants tiers

L'intégration de certaines fonctionnalités a nécessité l'emploi de composants tiers. En effet, certaines fonctionnalités nécessitent une présentation ou ont un fonctionnement spécifique. Ces composants sont des frameworks et des scripts javascript, mais aussi des bibliothèques de contrôles ASP.net.

Avant l'intégration définitive d'un composant tiers, une maquette montrant son fonctionnement était mise au point, puis présentée au maître d'ouvrage. Après approbation de ce dernier l'outil était utilisé, sinon un nouvel outil était proposé. L'utilisation de composant Javascript concerne notamment, la présentation d'images réduction et agrandissement, ainsi que la gestion de fenêtres de type popup ou modale. Pour cela j'ai utilisé les composants Highslide, Jcarousel, Lightbox, et Reflection.

Les composants ASP.net sont notamment FckEditor pour avoir un éditeur WYSIWYG puissant et éprouvé et MetaBuilders. MetaBuilders fournit notamment des composants tels que, la CheckedListBox et le MultiFileUpload. L'utilisation de ces derniers, est justifiée par le fait qu'ils ne sont pas présents dans les composants asp.net de base, de plus ils permettent de diminuer le temps de livraison de la solution.

Le chargement des composants JavaScript se fait en chargeant la zone de contenu Headers. De même l'utilisation des composants asp.net a été effectuée en utilisant les directives prévues à cet effet.

```
<asp:Content ID="Content1" ContentPlaceHolderID="Headers" runat="Server">
    <script type="text/javascript"
src="../../Fichiers/Javascript/lightbox/prototype.js"></script>
    <script type="text/javascript"
src="../../Fichiers/Javascript/lightbox/scriptaculous.js?load=effects"></scrip
t>
    <script type="text/javascript"
src="../../Fichiers/Javascript/lightbox/lightbox.js"></script>
    <script type="text/javascript"
src="../../Fichiers/Javascript/highslide/highslide-with-html.js"></script>
    <link rel="stylesheet" type="text/css"
href="../../Fichiers/Javascript/highslide/highslide.css" />
    <script type="text/javascript">
        hs.graphicsDir = '../../Fichiers/Javascript/highslide/graphics/';
        hs.outlineType = 'rounded-white';
        hs.numberOfImagesToPreload = 0;
        hs.showCredits = false;
        hs.lang = {
            loadingText: 'Chargement en cours...',
            fullExpandTitle: 'Clique ici pour agrandir',
            restoreTitle: 'Clique ici pour fermer',
            focusTitle: 'Titre',
            loadingTitle: 'Annuler'
```

```
};
</script>
</asp:Content>

<%@ Register Assembly="FredCK.FCKeditorV2" Namespace="FredCK.FCKeditorV2"
TagPrefix="fck" %>
<%@ Register Assembly="MetaBuilders.WebControls"
Namespace="MetaBuilders.WebControls"
TagPrefix="mwc" %>
```

Figure 18 : MetaBuilders checklistbox

Coloris

- ☐ Bleu
- ☐ Jaune
- ☐ Noir
- ☐ Rouge
- ☐ Vert

Figure 19 : Fckeditor pour la mise en forme de contenu

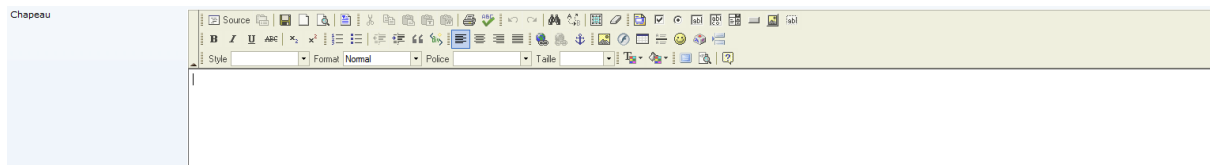


Figure 20 : MetaBuilders multifileupload

Parcourir...

--- Uploaded Files --- Remove

Figure 21 : Utilisation de jcarousel

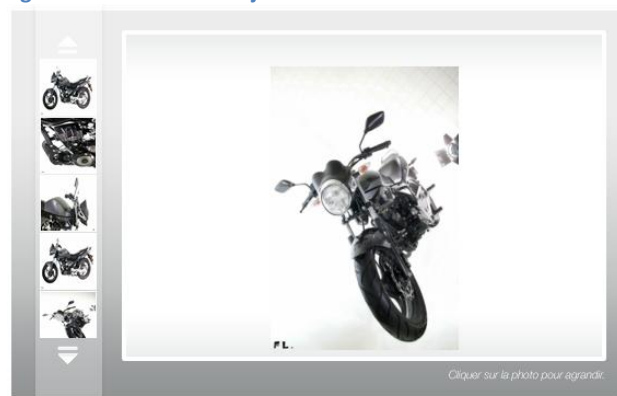


Figure 22 : Utilisation de lightbox



Figure 23 : Utilisation de reflection



Figure 24 : Utilisation de highslide



The screenshot shows a window titled "Liste des clients" with a close button (X) in the top right corner. Inside the window is a table with the following data:

Reference	Nom	Prenom	Ville	Email	
1Al20091201	Albina	Teddy	Bezons	teddyalbina@hotmail.fr	Sélectionner
1Pa20091201	Palisse	Cindy	Bezons		Sélectionner

There is a small icon in the bottom right corner of the window.

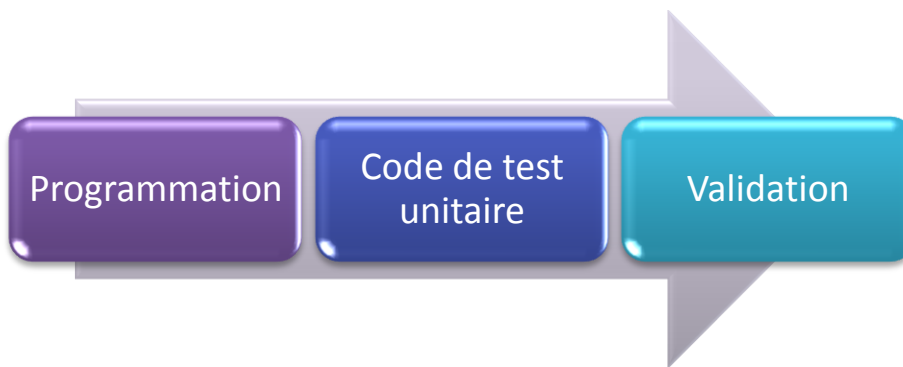
Références

- Highslide : <http://highslide.com>
- Jcarousel : <http://sorgalla.com/projects/jcarousel>
- Lightbox : <http://www.huddletogether.com/projects/lightbox2>
- Reflection : <http://cow.neondragon.net/stuff/reflection>

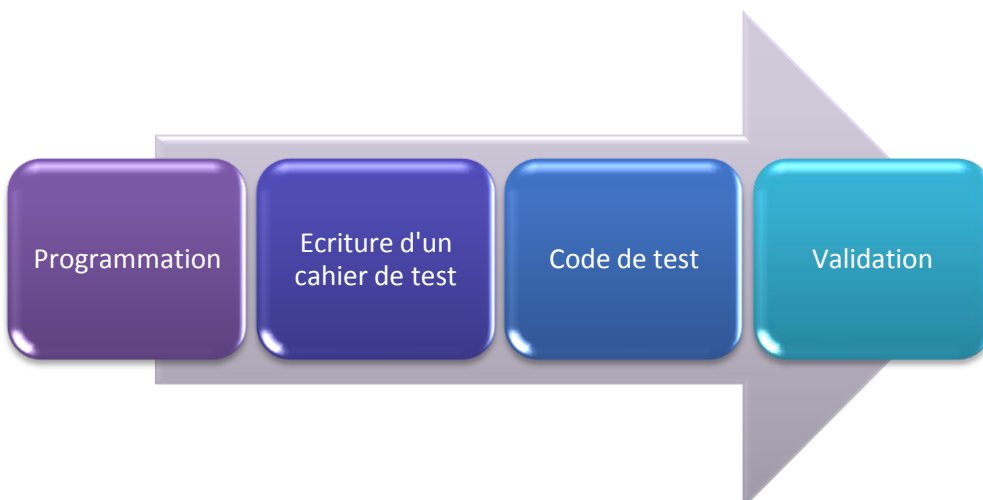
7. Test et validation des développements

Pour apporter une qualité optimale chaque développement a été validé, par une série de tests unitaires, par des tests manuels, ou encore par des tests fonctionnels en fonction du contexte de développement.

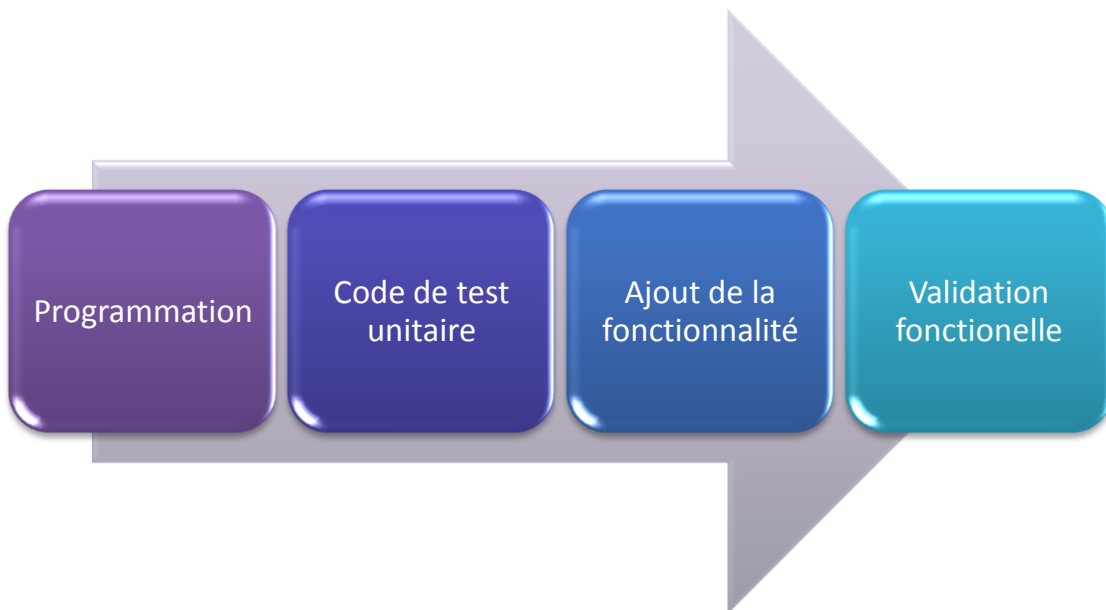
Pour les développements portant sur du code testable indépendamment de l'interaction avec une action extérieure, j'ai procédé à l'écriture de test unitaire. Une fois la batterie de tests passée, le code est alors ajouté à la solution.



Pour la validation du code du caddie, j'ai écrit et calculé manuellement une série d'opérations afin de prouver le bon fonctionnement des algorithmes, ainsi que les différentes erreurs de calculs pouvant survenir. Puis j'ai créé une application console dans laquelle j'ai écrit le code de test du caddie, j'ai ensuite vérifié que chacun des résultats renvoyé par le caddie correspondaient à ceux que j'avais calculé moi même. En outre, j'ai soumis les algorithmes du caddie à une série de données erronées. Une fois le fonctionnement du caddie validé il a été ajouté à la solution.



Les fonctionnalités utilisables depuis l'interface graphique étaient soumises en plus des tests unitaires à des tests fonctionnels. Ces tests étaient effectués par le maître d'ouvrage et les utilisateurs de la solution. Ils essayaient en situation réelle la nouvelle fonctionnalité, et me faisaient un retour d'informations, si le maître d'ouvrage validait la nouvelle fonctionnalité elle était alors ajoutée à la branche principale du projet. Lorsque ce n'était pas le cas, nous concertions afin de voir les points à améliorer. Ce cycle recommençait autant de fois que nécessaire.



Le test et la validation systématique de chaque développement prend un certain temps mais apporte plus de sécurité concernant la fiabilité du système. Néanmoins, il est apparu que les tests ne peuvent prémunir intégralement des bugs.

8. Déploiement

1. Choix de l'hébergeur

Mon chef de projet m'a confié le choix du prestataire qui assurerait l'hébergement du site web. J'ai commencé par dresser un tableau des offres présentées chez les principaux acteurs du marché, je suis aussi entré en contact téléphonique avec les équipes commerciales et techniques de certains hébergeurs, afin d'obtenir plus de précisions sur les offres.

Pour effectuer la sélection des offres je me suis appuyé sur les critères suivants :

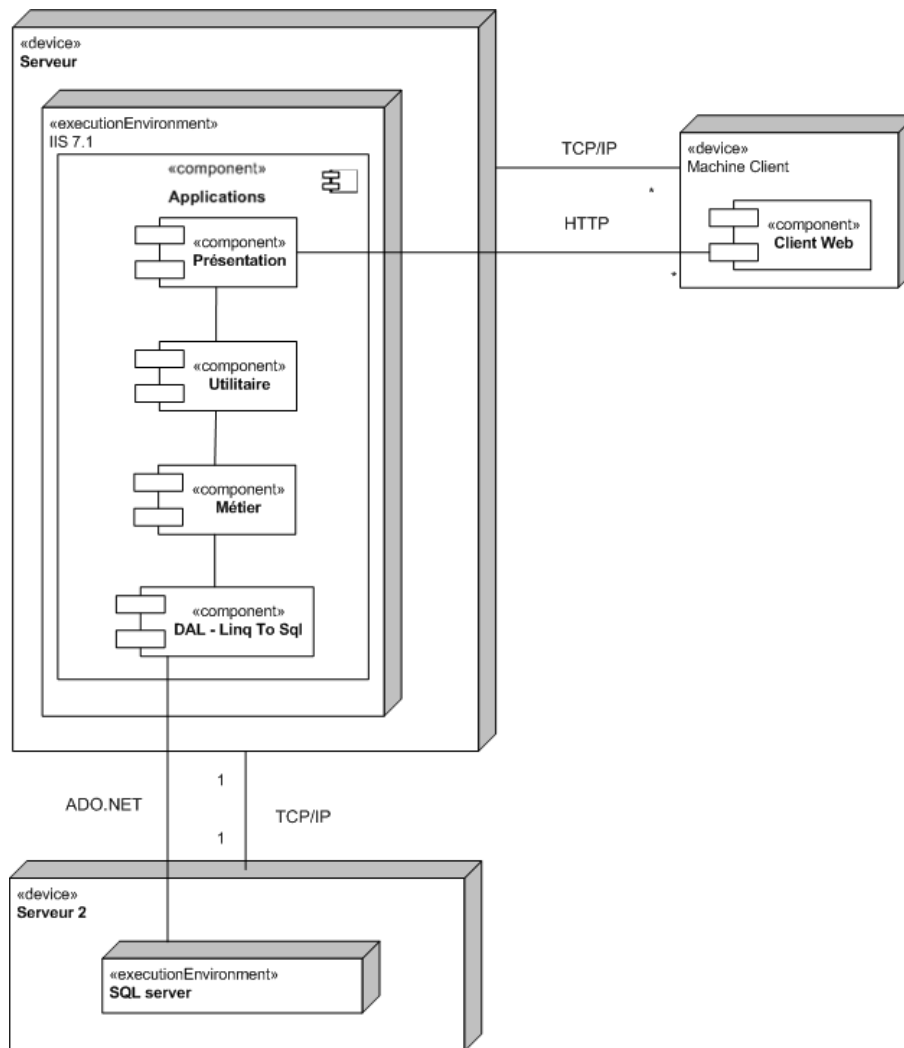
- Système d'exploitation : Windows server 2003 ou 2008
- Base de données : SQL server 2005 ou 2008
- Serveur web : IIS 6.1 ou IIS 7.0/7.5
- Espace disque : Minimum 1Go
- Langages : Asp.net 3.5
- Moteur application : .Net Framework 3.5 sp1
- Evolutivité : Possibilité de passer à une offre supérieure sans coupure de service
- Options : Présence d'un webmail, support du SSL, redirection email, accès à la base de données depuis l'extérieur, accès mail depuis l'extérieur, liste de diffusion,
- DNS : Possibilité d'utiliser un service externe (ici Gandi)
- Trafic : au moins 20go

J'ai effectué un comparatif des différentes offres (Annexe 10) que j'ai présenté au maître d'ouvrage, après concertation et prise de contact avec les services techniques de la société Ikoula. Il a été décidé de souscrire à son offre d'hébergement mutualisé Ikoula Deluxe.

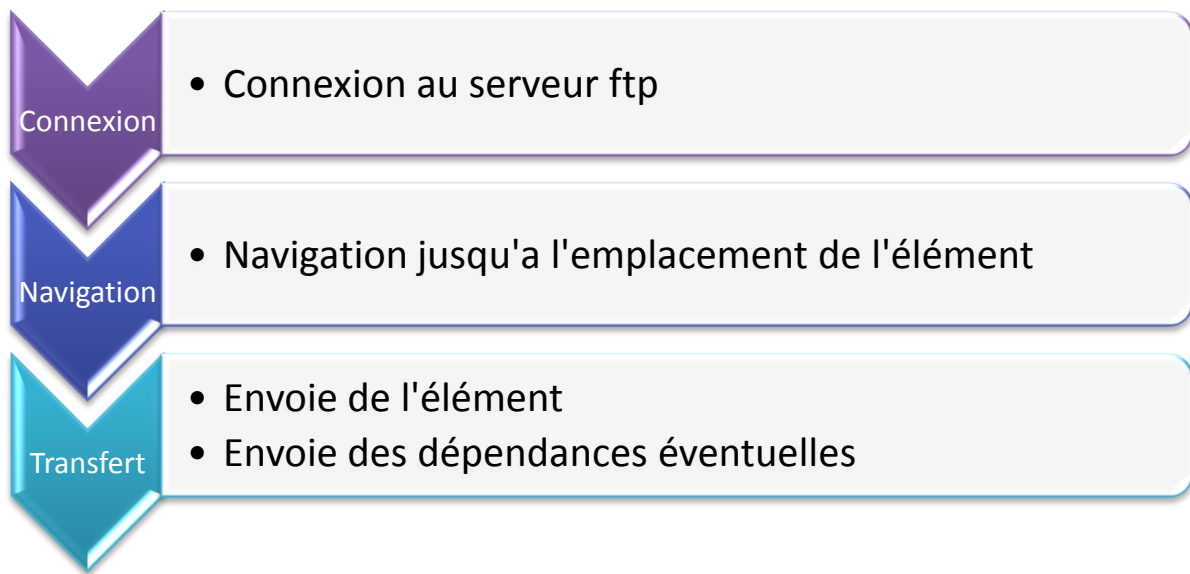
2. Mise en place de la solution

Le déploiement de la solution se fait par ftp, cette solution a été retenue car elle est éprouvée. De plus, un déploiement par ftp assure une meilleure gestion des éléments envoyés sur le serveur. Avant de déployer la solution j'ai modélisé le diagramme de déploiement de la solution dans un environnement type chez un hébergeur, afin d'avoir une vue concrète et simple de l'environnement de la solution.

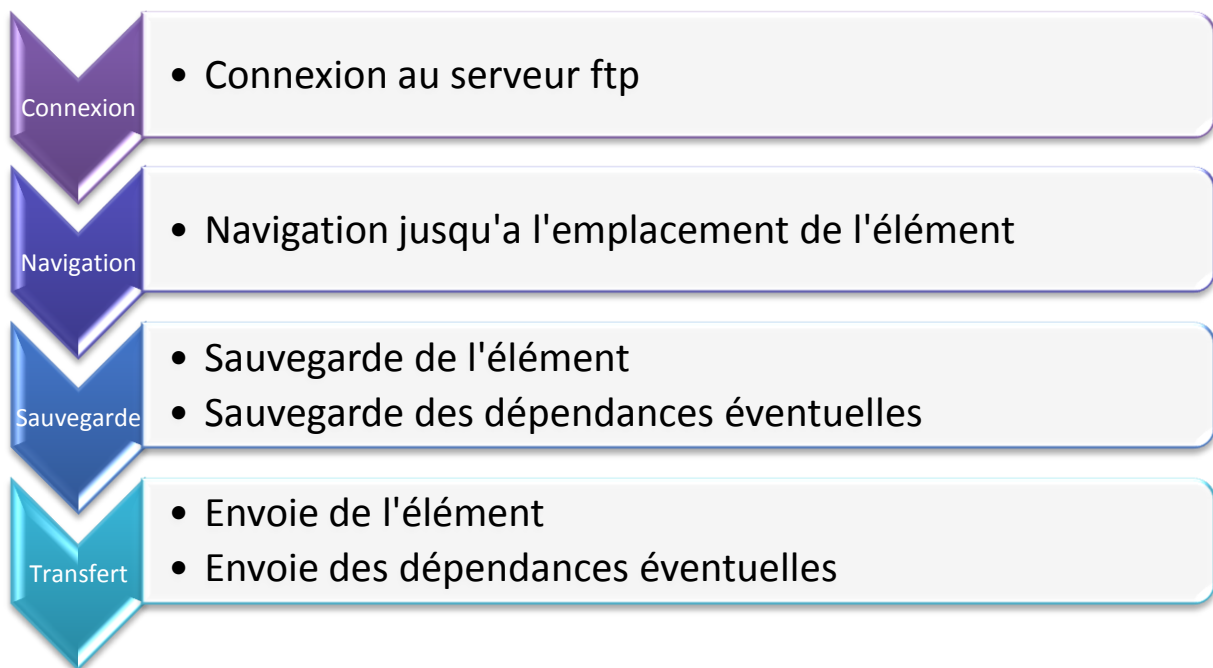
Figure 25: Diagramme de déploiement (cf : RTN-DC-V1)



La mise en ligne d'un nouvel élément se fait en plusieurs étapes :



Pour la mise à jour d'un élément la procédure à suivre est la suivante :



Cette procédure permet de garantir qu'en cas d'avarie il est possible de revenir en arrière en transférant sur le serveur de production la version précédente de l'élément mis à jour. L'ancienne version de l'élément n'est supprimée qu'après avoir validé le bon fonctionnement de la nouvelle version sur le serveur de production.

3. Configuration de la solution

La configuration de la solution se fait au travers du fichier web.config, la section appSetting permet de configurer un ensemble de variable à l'intérieur de la solution.

Configuration 1 : Configuration du seuil de disponibilité d'article

```
<add key="SeuilDisponibilite" value="0" />
```

Configuration 2 : Configuration des adresses e-mail

```
<!-- Adresse e-mail de l'émetteur des messages de suggestion -->
<add key="EmailSuggestion" value="no-reply@routeinn.com" />
<!-- Adresse e-mail de contact -->
<add key="EmailContact" value="contact@drim-moto.com" />
<!-- Adresse e-mail pour les réservations -->
<add key="EmailReservation" value="contact@drim-moto.com" />
<!-- Adresse e-mail en cas d'erreur -->
<add key="EmailErreur" value="bug@routeinn.com" />
```

La configuration de la chaîne de connexion à la base de données se trouve dans la section connectionStrings.

Configuration 3 : Configuration de la connexion à la base de données

```
<add name="RouteInn.DAL.Properties.Settings.
db_routeinn_finaleConnectionString" connectionString="Data
Source=localhost;Initial Catalog=db_routeinn_finale;User ID=sa;
Password=xbfkp1" />
```

La section system.web permet de charger les composants de la solution, et de configurer le serveur d'application.

Configuration 4 : Chargement d'un composant de l'application

```
<add tagPrefix="cc6" namespace="RouteInn.Controls"
assembly="RouteInn.Controls, Culture=neutral" />
```

Configuration 5 : Configuration du rendu xhtml, de l'encodage, et du runtime http

```
<xhtmlConformance mode="Strict" />
<globalization responseHeaderEncoding="utf-8" />
<!-- Configuration de la taille des fichiers pouvant être uploadé -->
<httpRuntime executionTimeout="600" maxRequestLength="409600" />
```

6/ Changements en cours de projet

De nombreux changements ont eu lieu en cours de projet, certaines fonctionnalités telles que la vente en ligne et la lettre d'information ont été temporairement annulées, ce qui a entraîné le non développement des fonctionnalités liées. Ces fonctionnalités sont :

- La visualisation de l'historique des commandes
- Le suivi de commande
- La possibilité pour un client de télécharger ses anciennes factures
- L'abonnement et le désabonnement de la lettre d'information

En outre, certaines sections du site n'ont pas été mise en ligne (Accessoires, Professionnel), tandis que d'autres sont apparues (Concessionnaires, Les échos de DRIM, Occasions-reprises, Liens). De plus, la section pièces détachées a changé d'objectif passant d'une section de présentation du catalogue de pièces détachées avec des fonctionnalités de commande en ligne, à une page présentant deux catalogues sous forme de fichiers PDF. La page caddie a été transformée en formulaire de contact afin que les clients qui souhaitent acquérir une moto puissent nous contacter. De même, le bouton "acheter" présent sur la page de présentation moto a été remplacé par un bouton "réserver".

Les parties concessionnaires et occasions-reprises ont entraînés le développement de nouvelles fonctionnalités, la mise à jour de la base de données, et des documents techniques. La page Lien est une page de type annuaire, sur laquelle figure les liens des sites sur lesquelles notre site a été enregistré.

Conclusion

Actuellement le site web `drim-moto.com` reçoit environs 617 visites par jour. Il a permis à l'entreprise de développer sa communication en rendant ses offres visibles tant aux particuliers qu'aux professionnels. De plus, grâce à cela Route Inn' Moto a étoffé son portefeuille clients. Par ailleurs, l'augmentation du nombre de clients professionnels à abouti à la création d'un réseau de concessionnaires, et sur la création d'une section dédiée sur le site.

Cette année de professionnalisation, m'a permis de mieux appréhender le métier de développeur. Ce métier ne se limite pas à la seule conception d'application, mais est composé de nombreuses facettes. En effet, il repose tant sur les qualités techniques que sur les qualités d'écoutes et d'observation. Ces qualités sont très importantes, car elles permettent de mieux cerner l'environnement dans lequel les applicatifs sont amenés à évoluer, mais aussi comment les acteurs tant au niveau individuel que collectif exercent leur métier. Elles permettent donc une meilleure adéquation de l'outil avec le métier de ses utilisateurs.

Le projet qui m'a été confié a été passionnant, car j'ai pu apprendre comment est organisé un projet d'entreprise. En outre, j'ai pu mettre en œuvre les méthodologies MERISE et UML qui sont des standards reconnus dans le génie logiciel. Par ailleurs, j'ai pu approfondir mes connaissances en C# et ASP.net qui sont de puissants outils pour la conception de site internet.

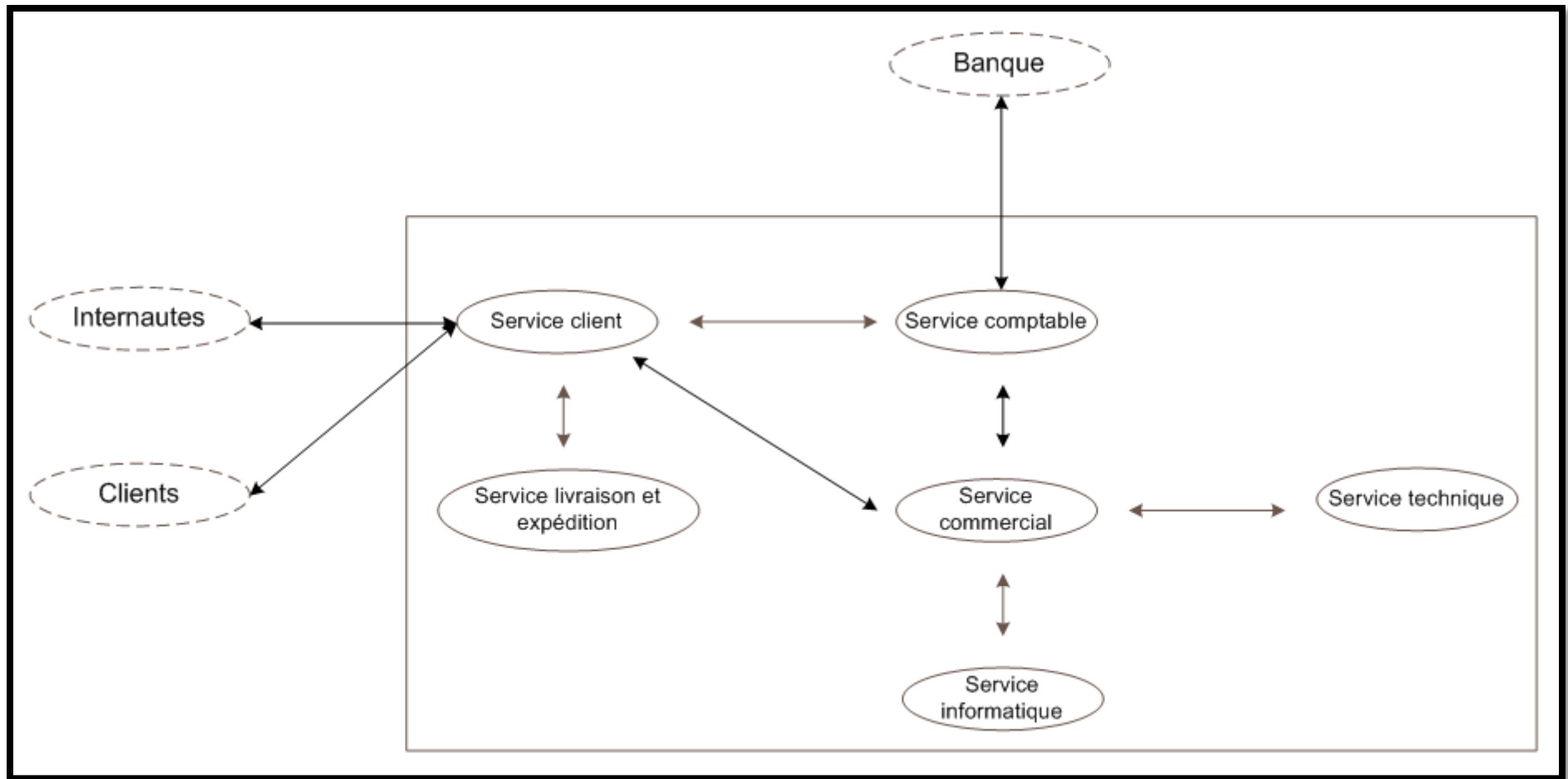
Bibliographie

Lecomte S. (2004) MySQL 5.0, 83-100.

Charroux B., & Osmani A., T-Mieg Y. (2008) UML2. *Pratique de la modélisation* : Vol. 2.

ANNEXE

Annexe 1: Diagramme conceptuel de flux



Annexe 2 : Matrice de flux entre les clients et les services de l'entreprise lors d'une commande

	Client	Service client	Service comptable	Service livraison et expédition	Banque
Client		Passe une commande			Paiement de la commande
Service client	Confirmation/Rejet de la commande		Vérification du paiement	Emission d'un ordre de traitement de la commande	
Service comptable		Validation/Invalidation du paiement			Vérification du paiement
Service livraison et expédition	Notification de la date de livraison ou d'expédition	Notifie l'expédition ou la livraison de la commande			
Banque			Notification de l'état paiement		

Emetteur : Horizontal

Récepteur : Vertical

Annexe 3 : Matrice de flux entre le client déposant un article en dépôt vente ou en reprise

	Client	Service commercial	Service technique	Service informatique
Client		Déclaration		
Service commercial	Emission d'une déclaration		Vérification de l'état du véhicule	Mise en ligne de l'article
Service technique		Estimation de l'état du véhicule et du prix		
Service informatique		Confirmation de la mise en ligne de l'article		

Annexe 4 : Matrice de flux entre les internautes et le service client

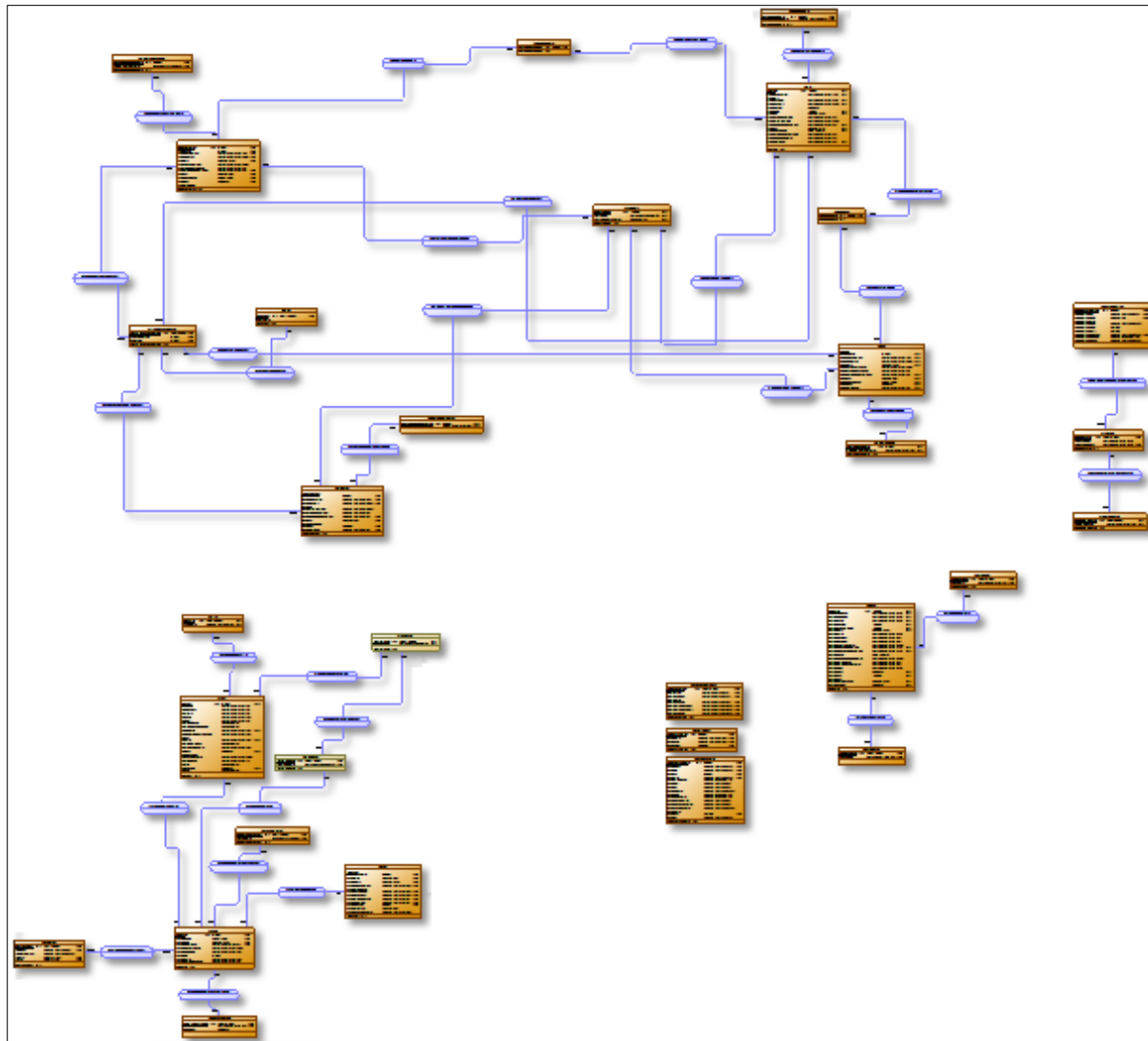
	Client	Service client
Client		Demande de renseignements
Service client	Envoie de renseignements	

Emetteur : Horizontal

Récepteur : Vertical

Annexe 5 : Matrice de flux entre les services comptable, commercial et client

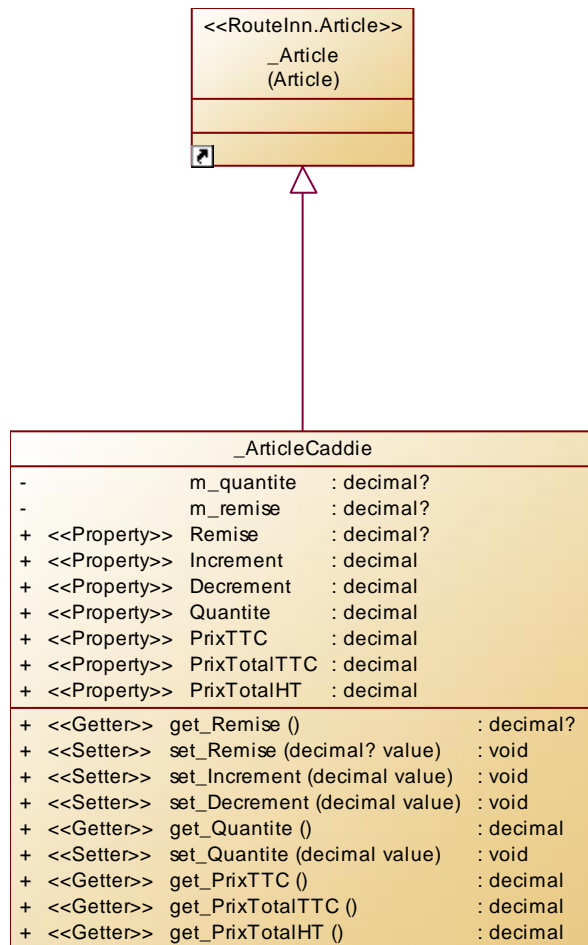
	Service comptable	Service commerciale	Service client
Service comptable		Indicateurs sur les ventes	
Service commerciale	Envoie de facture et devis		Envoie de fichier commerciale
Service client		Demande de fiche commerciale	



Annexe 8 : Diagramme de classe Caddie

Caddie			
-	m_article	: List<_ArticleCaddie>	= new List<_ArticleCaddie>()
-	m_TransporteurPrice	: decimal	= 0.0M
-	m_TransporteurTva	: decimal	= 0.0M
-	m_TransporteurId	: short	= -1
+	<<Property>> TotalHT	: decimal	
+	<<Property>> TotalTTC	: decimal	
+	<<Property>> TransporteurId	: short	
+	<<Property>> TotalHTAvecTransport	: decimal	
+	<<Property>> TotalTTCAvecTransport	: decimal	
+	<<Property>> TransporteurPrix	: decimal	
+	<<Property>> TransporteurTva	: decimal	
+	<<Property>> TransporteurTTC	: decimal	
+	<<Property>> ListeArticles	: IEnumerable<_ArticleCaddie>	
+	<<Property>> NombreArticles	: int	
+	<<Constructor>> Caddie ()		
+	Flush ()		: void
+	Add (int id_article, int id_couleur, string couleur, TypeArticle typeArticle, decimal prix, decimal tva, string designation, string categorie, string reference)		: void
+	Increment (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: void
+	Increment (int id_article, TypeArticle typeArticle)		: void
+	Decrement (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: void
+	Decrement (int id_article, TypeArticle typeArticle)		: void
+	Discount (decimal remise, int id_article, TypeArticle typeArticle)		: void
+	Delete (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: void
+	Delete (int id_article, TypeArticle typeArticle)		: void
+	<<Getter>> get_TotalHT ()		: decimal
+	<<Getter>> get_TotalTTC ()		: decimal
+	<<Getter>> get_TransporteurId ()		: short
+	<<Setter>> set_TransporteurId (short value)		: void
+	<<Getter>> get_TotalHTAvecTransport ()		: decimal
+	<<Getter>> get_TotalTTCAvecTransport ()		: decimal
+	<<Getter>> get_TransporteurPrix ()		: decimal
+	<<Setter>> set_TransporteurPrix (decimal value)		: void
+	<<Getter>> get_TransporteurTva ()		: decimal
+	<<Setter>> set_TransporteurTva (decimal value)		: void
+	<<Getter>> get_TransporteurTTC ()		: decimal
+	<<Getter>> get_ListeArticles ()		: IEnumerable<_ArticleCaddie>
+	<<Setter>> set_ListeArticles (IEnumerable<_ArticleCaddie> value)		: void
+	<<Getter>> get_NombreArticles ()		: int
-	FindArticle (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: int
-	FindArticle (int id_article, TypeArticle typeArticle)		: int
-	FindArticleIndex (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: int
-	FindArticleIndex (int id_article, TypeArticle typeArticle)		: int
-	Filter (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: void
-	Filter (int id_article, TypeArticle typeArticle)		: void
-	CheckArticle (int id_article, int id_couleur, TypeArticle typeArticle, string categorie)		: bool

Annexe 9 : Diagramme de la classe _ArticleCaddie



Annexe 10 : Tableaux de comparaison des offres d'hébergement

1&1	
Offre	Serveur privé virtuel Windows XL
Prix	29.99€ Ht/mois , 0€ les 3 premiers mois
Mémoire	1go
Hdd	20go
Os	W2k3
Bande passante	100 Mb/s garantie (symétrique)
Trafic/mois	illimité
Certificat SSL	Oui
SGBD	Mysql
Tse	?
Gestion Dns	Oui
Garantie hardware	Oui
Mise en service	9.99€ HT
Durée mini d'engagement	12 mois
Premières facture	11.95€ TTC
Alias	1000
Mail	illimité

Amen	
Offre	Serveur privé virtuel Windows
Prix	23.92€ TTC/mois
Mémoire	512
Hdd	1 à 4go (4go = +15€)
Os	W2k3
Bande passante	Inconnu
Trafic/mois	illimité
Certificat SSL	Oui
SGBD	Mysql / Sql server
Tse	Oui
Gestion Dns	Oui
Garantie hardware	Oui
Mise en service	9.99€ HT
Durée mini d'engagement	12 mois
Premières facture	11.95€ TTC
Ftp privé	Oui
Alias	Illimités
Mail	illimité
Mise en service	1h après paiement
Engagement 12 mois	(-33.33%)
Montant du pack	360€ HT (30 € /mois)
Montant à régler	430.56€ TTC

Ikoula	
Offre	Hébergement Windows
Prix	44€ TTC/mois
Mémoire	512
Hdd	2Go
Os	W2k8 / IIS7
Bande passante	Inconnu
Trafic/mois	180Go
Certificat SSL	Oui
SGBD	Mysql / Sql server
Tse	Oui
Gestion Dns	Oui
Garantie hardware	Oui
Mise en service	9.99€ HT
Durée mini d'engagement	12 mois
Première facture	11.95€ TTC
Ftp privé	Oui
Langages/Logiciels	Silverlight, Ajax 1.0, Asp.net 1.1 - 2.0 - 3.5, Asp 6.0, Php 4.3.4 / 5, Extensions Frontpage 98/00/02, SSI, CGI-BIN, Python, C, Perl JAVA, Tomcat,Wap
Base de données	Access,Foxpro, Paradox
	Mysql : Taille 1Go
	Mssql : Taille 250mo
Composant ASP	W3JMAIL
	ASPHttp
	ASP SmartUpload