

Introduction

En tant que fêru de nouvelles technologies, j'ai recherché un stage dans une entreprise et surtout un secteur qui saurait me combler. Plusieurs types de PFE¹ pouvaient convenir à mes attentes et je souhaitais élargir mes horizons et continuer d'agrandir mon bagage technique. Ayant une attirance plus particulière pour les technologies du Web², j'ai accepté la proposition de Smile Sud-Ouest en tant que développeur d'applications Web. En effet une SSII³ offre la possibilité d'acquérir beaucoup de connaissances rapidement et d'engranger de l'expérience à travers des projets divers et variés.

L'ensemble des technologies gravitant autour du Web étant en constantes évolution je savais que ce stage était pour moi l'occasion d'exercer dans un environnement qui me passionne. De plus, dès les entretiens j'ai ressenti la bonne ambiance et la convivialité dans cette agence, ceci a grandement participé à asseoir ce choix de PFE. Le fait que Smile soit une entreprise de taille conséquente a aussi contribué dans mon choix car ayant réalisé mon stage de troisième année dans une PME⁴ comptant seulement 4 employés, j'avais à coeur de découvrir ce qu'était un grand groupe.

Ce rapport s'efforcera au mieux de donner à son lecteur une idée précise de mes missions au sein de Smile pour les six mois de stage. Dans une première partie, je présenterai le groupe ainsi que sa position dans le secteur. Je porterai une attention particulière à l'agence Bordeaux où j'ai travaillé ainsi qu'aux diverses missions qui m'ont été proposées. Dans un second temps, j'aborderai les différents projets auxquels j'ai participé, leurs réalisations ainsi que les difficultés rencontrées et leurs solutions. Finalement, la dernière partie fera le bilan de ces six mois de stage et s'ouvrira sur les futurs projets en développement Web de l'entreprise.

1. Projet de Fin d'Etude

2. Abréviation pour World Wide Web

3. Société de Services en Ingénierie Informatique, maintenant appelé Entreprise de Services du Numérique (ESN)

4. Petite et Moyenne Entreprise

Remerciements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam pharetra sem neque, at lacinia tellus rhoncus sed. Aliquam erat volutpat. Sed in facilisis dolor. Nam gravida purus vel ante auctor, non venenatis mauris scelerisque. Maecenas odio nunc, dapibus eu varius ac, dapibus non diam. Nam a ligula sit amet velit porta auctor. Vestibulum vel luctus eros. Sed vestibulum neque tristique, fringilla felis sed, pharetra dui. Nulla facilisi. Aliquam viverra velit nulla, ut placerat purus auctor vitae. Aenean vitae mauris tristique metus placerat ultrices vel ac magna. Vestibulum massa leo, faucibus nec tincidunt vitae, bibendum vel tortor. Aliquam erat volutpat.

Table des matières

Introduction	1
Remerciements	2
1 Présentation de l'entreprise	5
1.1 Contexte	5
1.2 Groupe Smile	5
1.3 Agence de Bordeaux	6
1.4 Mes missions au sein de Smile	7
2 Développement d'applications Web à travers différents projets	9
2.1 Dediétrich thermique	9
Présentation du projet et état de l'art	9
Mes objectifs	9
Mes réalisations	9
Bilan du projet	11
2.2 Banque Française Mutualiste (BFM)	12
Présentation du projet et état de l'art	12
Mes objectifs	12
Mes réalisations	13
Bilan du projet	19
2.3 POC (Proof Of Concept) thème responsive Gifi	21
Présentation du projet et état de l'art	21
Mes objectifs	21
Mes réalisations	21
Bilan du projet	22
2.4 Refonte graphique Darjeeling	24
Présentation du projet et état de l'art	24
Mes objectifs	24
Mes réalisations	24
Bilan du projet	30
2.5 Snowleader refonte 2015	31
Présentation du projet et état de l'art	31
Mes objectifs	31
Mes réalisations	31
Bilan du projet	37
Conclusion	39
Enseignements et bénéfices	39
Les projets à venir	39

Annexes	40
Annexe Code CSS simulateur d'épargne	40
Annexe Transmission de données d'un formulaire	42
Annexe Illustrations pages téléconseillers	43
Annexe Illustration header Darjeeling	45
Annexe Installateur Magento pour ajouter un attribut au BO	46
Annexe Script jQuery d'ajustement d'une info bulle	47
Annexe Illustration voir plus de produits Darjeeling	48
Annexe Illustration dashboard Snowleader	49
Annexe Condition sur le lien d'impression de facture	50
Annexe Illustration back-office client Magento	51
Annexe Déclaration du controller custom loyalty	52
Annexe Illustration formulaire de modification des informations personnelles	55
Annexe Architecture logicielle MVC (Modèle Vue Contrôleur	56
Table des illustrations	58

1 Présentation de l'entreprise

1.1 Contexte

Le World Wide Web, littéralement la « toile (d'araignée) mondiale », communément appelé le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. Des sites il y en avait un seul en août 1991, en 2000 nous en comptons déjà 10 000 000, aujourd'hui c'est plus de 947 000 000. Ce chiffre exorbitant ne cesse d'augmenter de jour en jour et le nombre d'entreprises concevant ces sites augmentent lui aussi inexorablement. Aujourd'hui nous faisons face à de nouvelles problématiques dans le domaine de la création d'applications Web ou sites Web, à savoir une complexité des fonctionnalités demandées, une compatibilité cross-plateformes (PC de bureau, tablette et mobile) mais aussi avec toutes les versions des navigateurs (IE8, IE9, Chrome, Firefox, Safari,...) et des systèmes d'exploitations (Windows, MacOS, Linux, iOS, Android,...) et enfin des performances, en terme de vitesse de chargement des pages, toujours plus rapide.

1.2 Groupe Smile

Dans ce contexte, Smile est une société d'experts des architectures web et des solutions open source fondée en 1991. Cependant ce n'est qu'à partir de 2001 que Smile commence à construire son expertise des solutions open source¹ : un choix d'avenir que beaucoup de ses concurrents n'osent pas alors entreprendre, mais qui s'avèrera être un choix payant. En France et dans le monde ce sont plus de 700 collaborateurs qui opèrent dans une variété de domaines, Smile étant le premier intégrateur européen de logiciel libre. Dans l'hexagone ce sont 8 agences (Paris, Lyon, Nantes, Bordeaux, Montpellier, Marseille, Lille et Grenoble) qui se partagent les différents projets, auxquels s'ajoutent les 8 autres locaux à l'étranger (Amsterdam, Bruxelles, Genève, Zurich, Casablanca, Kiev, Moscou et Abidjan), pour arriver à un groupe comptant 16 infrastructures à travers le monde.

Acteur engagé dans les progrès de l'Internet² depuis 1995, Smile a réalisé quelques-uns des plus grands sites de l'Internet français, des sites à forte valeur ajoutée et à forte audience. Smile a également été choisie par les plus grandes entreprises françaises pour concevoir, réaliser et maintenir des applicatifs Intranet stratégiques, servant des centaines d'utilisateurs sur des milliers de transactions. On notera par exemple quelques grands clients comme le Ministère de la Culture, la Société Générale, Eiffage Immobilier, Dassault Systèmes, Eurosport, Krys, Carrefour,...

On trouve chez Smile plusieurs corps de métiers :

Ingénierie Le pôle ingénierie accueille le cœur de métier de l'activité de Smile. Il réunit l'ensemble des équipes en charge de concevoir et de produire les applications web répondant aux besoins des clients.

1. Désigne un logiciel dans lequel le code source est à la disposition du grand public

2. Réseau informatique mondial accessible au public comportant des services variés comme le Web notamment

Commerce Le pôle commercial regroupe les trois formes d'activités mises en oeuvre par Smile dans sa recherche de business : la prospection, l'avant-vente et la fructification.

Consulting Le pôle consulting se caractérise par son ouverture et sa polyvalence sur les domaines suivants : Enterprise Content Management (gestion de contenus d'entreprise), Assistance à Maîtrise d'Ouvrage (fonctionnel), Conseil Technique, Business Intelligence (Décisionnel) ,...

Système Le pôle système travaille au cœur des Systèmes et Réseaux de Smile et de ses clients. L'équipe œuvre sur la maintenance et l'évolution du système interne de Smile, à tous les niveaux : serveurs, réseaux, messagerie, téléphonie,... Pour l'externe, l'équipe a pour fonction d'assurer l'expertise systèmes et réseaux de Smile (architecture système-réseaux, exploitation, hébergement,...) auprès de ses clients.

Agence Digitale L'agence Digitale est une équipe qui travaille pour les clients de Smile à la refonte ou à l'élaboration de leur identité visuelle : e- Marketing, charte graphique, accessibilité, interactivité et animation,...

Administratif Le pôle administratif recouvre des métiers aussi nombreux que variés : comptabilité, contrôle de gestion, finances mais aussi ressources humaines et recrutement ou encore communication et marketing.

Pour le pôle ingénierie qui nous intéresse plus particulièrement on trouve de nombreux métiers là aussi, je me contenterai de les citer sans les détailler un à un :

- Ingénieur études et développement (IED)
- Développeur
- Chef de projet
- Expert technique
- Chef de projet fonctionnel
- Chef de projet technique
- Consultant expert veille IT³
- Directeur de projet

Je reviendrai par la suite sur le rôle de l'IED puisque cela correspond à ma fonctions en tant que stagiaire au sein de Smile.

1.3 Agence de Bordeaux

A Bordeaux ce sont une quarantaine de collaborateurs, majoritairement ingénieurs, qui développent des applications Web utilisant les CMS⁴ open source (eZ Publish, Drupal), la solution de e-commerce Magento, mais aussi l'ensemble des solutions Open Source portées par Smile. Récemment le développement de projets Symfony 2⁵ est venu s'ajouter aux missions de l'agence de Bordeaux notamment du fait de l'importance que prend ce framework⁶

3. Technologies de l'information et de la communication

4. Content Management System, est un site web disposant de fonctionnalités de publication et offrant en particulier une interface d'administration permettant à un administrateur de site de créer ou organiser du contenu

5. Framework libre et français écrit en PHP 5 destiné majoritairement aux professionnels du développement permettant de faciliter et d'accélérer la création d'un site web

6. En programmation informatique, un framework ou structure logicielle est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel

PHP⁷ sur le marché.

Qui dit développement, dit langage informatique, à Bordeaux il y a majoritairement un langage utilisé à savoir le PHP, même si des projets en Java⁸, .Net⁹ ont déjà et sont encore réalisés. En plus de ceux-ci, sont utilisés au quotidien les langages basiques du Web à savoir HTML5¹⁰, CSS3¹¹, JavaScript¹² et toutes les variantes ou framework qui y sont liés (jQuery, Sass,...). Quand à l'environnement de travail, toutes les machines des développeurs sont équipées de Linux¹³ avec une distribution spécifique à Smile. Chacun est libre d'y installer, en plus des logiciels de bases fournis, tous les outils nécessaires à son travail de développement. Les locaux sont organisés en open-space afin de faciliter les échanges entre collaborateurs et un regroupement par projet est mis en place toujours dans le même but.

1.4 Mes missions au sein de Smile

Ma première mission au sein de Smile a été d'installer et de mettre en place mon poste de travail, cela est passé par l'installation du Linux custom sur ma machine ainsi que des logiciels nécessaires au développement. J'ai notamment utilisé PhpStorm comme IDE¹⁴ ainsi que le navigateur Chrome de Google pour déboguer mes pages WEB. En plus de cela j'ai eu besoin d'outils comme VirtualBox pour simuler des environnements Windows à des fins de test de compatibilité, ou encore Meld un logiciel de comparaison de fichier très utile pour détecter des erreurs dans le code. S'en est suivie une période d'autoformation sur les différentes technologies utilisées par l'agence, à savoir Drupal, Magento et EzPublish.

J'ai par la suite été placé sous la tutelle d'un chef de projet avec qui j'ai commencé le développement pour des clients sur des projets existants. Il s'agit de projets de Tierce maintenance applicative (TMA), les clients demandent des évolutions ou des corrections sur leurs sites et nous nous chargeons de les effectuer, de les tester puis de leur livrer et de vérifier avec eux que cela leur convient et répond à leurs attentes. Dans les missions qui m'étaient confiées j'avais plusieurs aspects du métier à mener, à savoir, la relation avec le client, l'appeler discuter avec lui de l'avancement et des fonctions souhaitées concernant sa demande. Mais aussi le développement en utilisant différents langages, différentes technologies et travailler sur des sites de nature et aux objectifs différents.

Dans tous les cas j'avais un gros travail d'autoformation, dû au fait que je travaillais sur des technologies que nous n'avions pas abordé en cours à l'INSA. En plus de ça, des formations internes étaient proposées et c'était un plus pour moi d'y participer afin de faire évoluer mon bagage technique et ainsi devenir encore plus polyvalent. En effet la polyvalence est aussi un objectif chez Smile car tous les développeurs doivent être capable de gérer le front-end¹⁵, le back-end¹⁶ et les aspects réseaux liés au projet. L'idée étant d'avoir des programmeurs full-stack¹⁷ et acquérir une grande adaptabilité, ce qui est un plus quand on

7. Langage de programmation libre principalement utilisé pour produire des pages Web dynamiques. PHP est un langage orienté objet et a permis de créer un grand nombre de sites web célèbres, comme Facebook, YouTube, Wikipedia, Google,... Il est aujourd'hui considéré comme la base de la création des sites Internet dits dynamiques

8. Langage de programmation informatique orienté objet détenu par la société Oracle

9. Ensemble de produits et de technologies informatiques de l'entreprise Microsoft pour rendre des applications facilement portables sur Internet

10. HyperText Markup Language 5, langage permettant notamment le développement d'applications Web

11. Cascading Style Sheets, langage qui décrit la présentation des documents HTML (couleur, mise en page,...)

12. Langage de programmation de scripts principalement employé dans les pages web interactives

13. Nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau Linux

14. Integrated Development Environment, un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels

15. Partie du site visible par les utilisateurs, il s'agit de l'interface entre l'utilisateur et le back-end

16. Partie du site non visible par les utilisateurs et s'exécutant côté serveur

17. Développeur maîtrisant les principales technologies et les principaux langages de programmation actuellement utilisés afin d'intervenir à la fois sur le front-end et le back-end des sites Internet

débute sur le marché de l'emploi.

Afin d'aplanir les connaissances des nouveaux arrivant chez Smile, l'entreprise a mis en place une formation interne, la Smile Academy. On y aborde notamment la gestion de version avec SVN, l'utilisation de Linux, le langage PHP, les bonnes pratiques du Web, ... Un ensemble de prérequis indispensable au bon fonctionnement des projets car on y voit par exemple les process propre à Smile dans la gestion des projets. Cette formation de trois jours a eu lieu dans nos locaux à Bordeaux avec les nouveaux arrivants, nous étions six personnes à suivre la formation. Selon moi il s'agit d'une bonne initiative de la part du groupe, qui plus est le formateur est une personne d'expérience chez Smile cela nous permet de poser bon nombres de questions que ce soit sur les aspects techniques ou humains du métier.

Afin de faciliter la compréhension du rapport je me dois de détailler et d'expliquer certains outils utilisés chez Smile. Je vais commencer avec l'outil de gestion appelé Gescom dont le but est de répertorier le travail effectué par les différents collaborateurs au sein de Smile. L'idée étant qu'à minima toutes les fins de semaine chaque collaborateur impute le temps passé sur chacun de ses projets afin de pouvoir suivre ce que l'on facture au client. Ceci est un outil interne auquel seuls les collaborateurs de Smile ont accès à la différence du suivant Redmine. Nous nous en servons pour répertorier les projets et c'est ici que le client nous fais parvenir ses demandes sous forme de tickets. Ceux-ci sont ensuite chiffrés par le chef de projet puis affectés aux différents développeurs qui travaillent dessus tout en indiquant au client le temps passé, le reste à faire et tout ce dont le client pourrait avoir besoin. Lorsque nous travaillons nous utilisons un système de virtualisation afin de simuler la présence d'une machine par projet. Ce système appelé LXC¹⁸ nous permet d'avoir un conteneur par projet et s'avère très utile dès que nous souhaitons partager notre environnement de travail à un autre collaborateur car il suffit de packer sa LXC puis de la donner à une autre personne, il se retrouvera ainsi avec un environnement de travail identique au notre. Comme la plupart des SSII Smile utilise un gestionnaire de version¹⁹, historiquement c'est SVN²⁰ qui était utilisé chez Smile et c'est celui-ci que j'ai pris en main dans les projets où je suis intervenu, mais il est à noter que Git²¹ fait son apparition sur de plus en plus de projets au sein du groupe.

18. Linux Containers est un système de virtualisation, il est utilisé pour faire fonctionner des environnements Linux isolés les uns des autres dans des conteneurs partageant le même noyau et une plus ou moins grande partie du système hôte. Le conteneur apporte une virtualisation de l'environnement d'exécution (Processeur, Mémoire vive, réseau, système de fichier...) et non pas de la machine. Pour cette raison, on parle de « conteneur » et non de machine virtuelle.

19. La gestion de versions consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers. Elle permet notamment d'accéder à l'historique des modifications (ce qui a été modifié, qui l'a modifié et quand la modification a eu lieu), de partager le code avec les différents contributeurs, de revenir à une ancienne version du fichier,...

20. Subversion est un logiciel de gestion de versions il fonctionne sur le mode client-serveur avec un serveur informatique centralisé et unique où se situent les fichiers constituant la référence (le « dépôt » ou « repository » en anglais)

21. Git est un logiciel de gestion de versions décentralisé consistant à voir l'outil de gestion de versions comme un outil permettant à chacun de travailler à son rythme, de façon désynchronisée des autres, puis d'offrir un moyen à ces développeurs de s'échanger leur travaux respectifs. De fait, il existe plusieurs dépôts pour un même logiciel

2 Développement d'applications Web à travers différents projets

2.1 Dedietch thermique

Présentation du projet et état de l'art

De Dietrich est une entreprise spécialisée dans l'électroménager, le ferroviaire et le chauffage. Dans le cadre de ce projet nous travaillons sur la partie chauffage de l'entreprise, le développement du site a été réalisé avec le CMS EzPublish et plusieurs versions de ce site sont déclinées. Nous y trouvons un site principal en français www.dedietch-thermique.fr, ainsi que des déclinaisons pour le marché russe, belge, espagnol,... A tout cela vient s'ajouter des sites mobiles eux mêmes ayant plusieurs versions en fonction des différents pays. Un service après-vente est aussi disponible en ligne et c'est dans ce cadre là que je suis intervenu. La demande du client consistait en la réalisation de sites mobiles pour leur service après-vente pour les marchés russe et belge. Ces sites existaient déjà en version française et tchèque, il fallait donc s'inspirer du travail qui avait été fait pour développer les nouveaux.

Mes objectifs

- Mettre en place deux nouveaux sites mobiles pour les marchés russe et belge avec une version français belge et une version néerlandais belge
- Découvrir et prendre en main le CMS EzPublish ainsi que les méthodes de travail et de livraison propres à Smile
- Corriger des bugs d'affichages dus aux différentes traductions
- S'assurer que les nouveaux sites créés soient compatibles avec toutes les fonctionnalités précédemment développées

Mes réalisations

Copie de l'arborescence J'ai commencé par copier l'arborescence des fichiers du site français déjà existant en utilisant un script développé par Smile afin de déclarer à EzPublish la création d'un nouveau site et de pouvoir l'administrer dans le back-office. J'ai ensuite modifié les fichiers de configuration du site afin d'indiquer les nouvelles URL correspondantes aux nouveaux sites ainsi que les paramètres de langues. La particularité étant la présence de plusieurs fichiers de configuration correspondant chacun à une étape du développement à savoir local, recette et production. Pour chaque cas le fichier de configuration doit correspondre au back-office de l'étape correspondante.

Pousser en recette Une fois mon travail réalisé et testé en local je l'ai envoyé en recette afin que le client puisse y avoir accès et puisse vérifier que cela lui convenait. Pour réaliser ceci, là aussi des scripts ont été développés par Smile facilitant cette tâche. Ceci étant fait il reste à avertir le client via Redmine que le développement est terminé et qu'il doit nous faire des retours en vue d'éventuelles modifications à apporter.

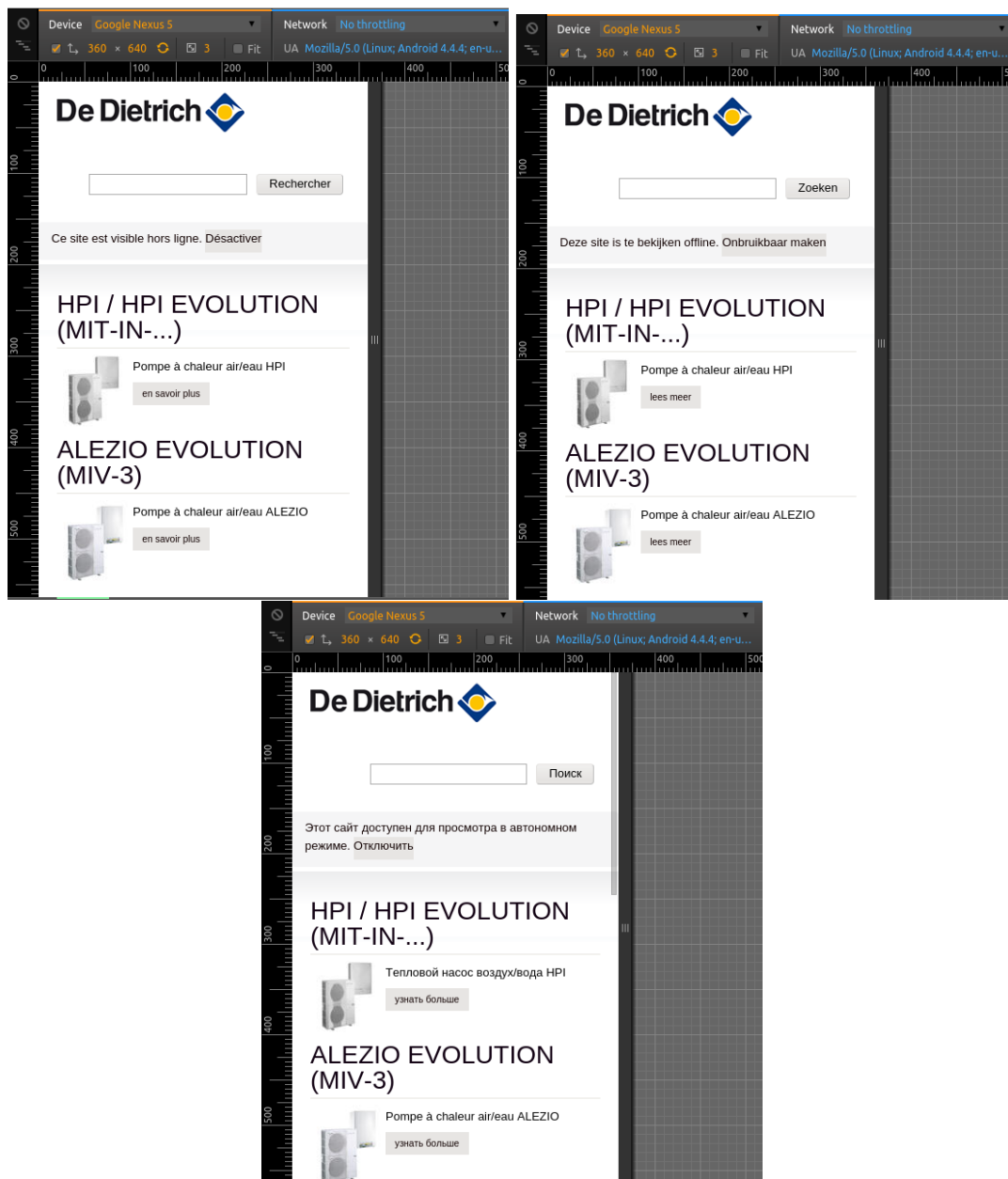


FIGURE 2.1 – Nouveaux sites mobiles (2 pour la Belgique et 1 pour la Russie)

Corriger les bugs Une grosse partie du travail consiste en la correction de bugs détecté par le client. J’ai notamment eu des modifications à apporter en terme d’affichage car les traductions dans les différents langages changent les tailles des éléments et créés des problèmes. Ceci consistait simplement en la modification des propriétés CSS pour adapter l’affichage. Des fichiers de traductions ont aussi dû être créé pour leur permettre de traduire les textes. Des fichiers pdf contenant les notices des produits ne se génèrent plus sur les nouveaux sites créés, il a donc fallu modifier le code PHP pour s’adapter à ses nouveaux sites.

Validation par le client et livraison en production Une fois que le client a validé le travail, je suis passé à la livraison en production de la même manière que pour la recette, en utilisant un script qui facilite la chose et qui prend juste en paramètre le numéro des commits¹ à livrer. Que ce soit pour la livraison en recette ou en production il faut aussi reporter les modifications effectuées sur le back-office local vers celui de recette et de production. Une fois la livraison effectuée j’ai vérifié le bon fonctionnement du code

1. Dans un système de gestion de version tel que SVN que nous utilisons chez Smile, la validation des modifications effectuées en local vers le serveur central est appelé un commit, qui s’accompagne généralement d’un commentaire pour expliquer les modifications apportées

mis en place et qu'aucun bug ne s'était glissé. Après le client peut toujours remonter un problème via Redmine, j'ai donc la responsabilité d'être à l'écoute au cas où un problème surgirait ou si une autre modification devait être apportée.

Bilan du projet

Cette première participation à un projet concret fut très enrichissante aussi bien sur le plan technique qu'humain. Du point de vue technique cela m'a permis d'acquérir des connaissances sur le CMS EzPublish que je ne connaissais pas. De plus l'utilisation exclusive du terminal Linux a consolidé mes bases dans ce domaine notamment grâce à la manipulation des scripts internes à Smile que j'ai eu l'occasion de modifier moi même afin de répondre à mes besoins. En ce qui concerne EzPublish j'ai eu globalement à faire à l'ensemble des aspects du CMS ce qui m'a donné une vue générale de son fonctionnement et me permettra d'aborder les prochains projets avec plus de facilités. Quand à mes réalisations, tout s'est bien déroulé, les sites sont maintenant disponibles en ligne et le client en est satisfait.

C'est un des avantages de ce métier, c'est que l'on peut directement voir l'aboutissement de son travail une fois que celui-ci est terminé. Pour ce qui est de l'aspect humain ce premier projet m'a enseigné les méthodes de gestion au sein de Smile, tout le cheminement pour passer de la demande du client à la livraison en production. Sans oublier l'utilisation des outils internes notamment Redmine afin d'informer le client de l'avancement du développement et des problèmes rencontrés. Ainsi que la communication entre développeurs en cas de problème ou simplement pour avoir de la visibilité sur l'état du projet. Malgré que j'ai passé peu de temps sur le projet (à peine un mois), celui-ci a été très formateur et m'a permis d'appréhender la suite avec plus de facilités. C'est donc naturellement que j'ai enchaîné avec un autre projet plus long et plus compliqué, où je pourrais mettre en pratique les connaissances acquises lors de cette première expérience, que je vais détailler par la suite.

2.2 Banque Française Mutualiste (BFM)

Présentation du projet et état de l'art

La Banque Française Mutualiste est une banque, créée en 1986 sous le nom de Banque Fédérale Mutuelle, par les mutuelles de la Fonction publique qui ont souhaité proposer aux agents du Secteur public une banque citoyenne, qui conjuguerait les services bancaires et les valeurs mutualistes. Dans le cadre de ce projet Smile à un contrat de TMA et l'objectif est la refonte du site existant.

Dans les tâches à effectuer il y a notamment la conversion de modules anciennement en Flash vers du HTML5, l'ajout de fonctionnalités et quelques retouches graphiques. Une des possibilités intéressantes est de pouvoir réaliser des simulations de prêts directement sur le site, cependant ceci est réalisé en Flash qui est une technologie propriétaire d'Adobe de moins en moins utilisée car non compatible avec les plateformes mobiles. Les clients souhaitent donc redévelopper ces simulateurs en HTML5/JavaScript. De plus ce type de simulateur est à intégrer sur d'autres sites partenaires ainsi que sur Facebook. Le problème de cette intégration est que BFM propose plusieurs types de prêts (moins de 35 ans, prêt auto, prêt travaux,...) et qu'ils souhaiteraient avoir un seul simulateur avec une liste déroulante permettant de choisir le type de prêt et que les taux s'adaptent en conséquence. Car si sur le site de BFM avoir un simulateur par page correspondante aux différents types de prêts ne posent pas de problème, il n'en est pas de même pour les sites partenaires où l'on souhaite tout intégrer dans un bloc de taille limitée (notamment sur Facebook).

Parmi les nouvelles fonctionnalités que demandaient le client on trouve un template vide leur permettant de créer des landing page², ils avaient l'habitude de les faire à part puis de les déposer sur le serveur, l'idée était d'intégrer ceci au CMS EzPublish pour faciliter l'élaboration et la maintenance de ces pages. Créer un nouveau paramètre que l'on peut ajouter dans les formulaires permettant la redirection vers la page d'un produit existant tout en gardant les données déjà saisies.

Il s'agit donc d'un projet complet et complexe avec de nombreuses tâches à effectuer même si certaines se basent sur de l'existant et qu'il s'agit simplement d'un changement de technologie. Une particularité de ce projet est que nous n'avons pas accès aux serveurs de production du fait que le client soit une banque, ceci nous oblige donc à développer en interne, à leur montrer via un serveur de recette auxquels ils ont accès mais pour les livraisons en production nous leur transmettons les fichiers et c'est eux mêmes qu'ils les déploient sur leur serveur. Le principal inconvénient étant que nous n'avons pas accès au code qui est en production ce qui est parfois problématique lorsque nous n'arrivons pas à reproduire les problèmes en local. Ceci oblige à beaucoup de patience et de précision lors des livraisons et des explications pour déployer le code que nous leur fournissons.

Mes objectifs

- Développer un simulateur de prêt en HTML5/JavaScript à intégrer sur différents sites partenaires (notamment une page Facebook), permettant la sélection du type de prêt dans une liste déroulante
- Convertir un simulateur d'épargne flash vers une version en HTML5/JavaScript
- Développer un template intégré à EzPublish pour la création de landing page et réaliser un exemple en s'inspirant de l'existant
- Développer un nouvel élément de formulaire permettant la redirection vers une page produit existant et transmettant les données déjà saisies par l'utilisateur

2. La landing page désigne la page sur laquelle arrive un internaute après avoir cliqué sur un lien (commercial, email, bandeau publicitaire,...). Elle a pour vocation d'être simple, belle et épurée et de transformer un simple internaute en client

- Développer une page Web à disposition des téléconseillers intégrant une page d'orientation vers les différentes simulation de prêts personnels et une fonctionnalité d'envoi de mail type intégrant un lien vers une simulation avec des paramètres intégrés dans l'URL

Mes réalisations

Développer un simulateur de prêt en HTML5/JavaScript à intégrer sur différents sites partenaires (notamment une page Facebook), permettant la sélection du type de prêt dans une liste déroulante

Pour coder cette fonctionnalité il s'agissait d'un mélange de JavaScript avec le framework jQuery et d'EzPublish. L'idée de la conception était de récupérer les différents simulateurs existant en fonction du choix de l'utilisateur dans la liste déroulante puis grâce à un appel AJAX de l'afficher. Plusieurs difficultés se présentaient à moi, récupérer le bon simulateur en fonction du choix de l'utilisateur, réussir à le charger rapidement (que ce soit transparent pour l'utilisateur) et enfin ajuster l'affichage car pour l'intégrer sur Facebook le client avait besoin du simulateur seul sans le design habituel du site. J'ai donc commencé par générer la liste déroulante en fonction des types de prêts choisis dans le back-office. En effet ce simulateur est configurable et s'adapte en fonction des types de prêts sélectionnés. Je récupère ensuite les URL correspondantes afin de pouvoir les utiliser par la suite lors de l'appel AJAX. En parallèle il a fallu créer dans le back-office la classe correspondante pour pouvoir contribuer le contenu à afficher sur les pages du CMS. C'est notamment ici que l'on décide des types de prêts qui seront proposés.

FIGURE 2.2 – Récupération des données depuis le back-office et affichage de la liste déroulante

```

1 <div class="content">
2   <select id="loan-choice">
3     {foreach $node.data_map.loan_choice.content.rows.sequential as $key => $row}
4
5       <option value="choice-{$key}" id="choice-{$key}">{$row.columns.0 | wash()}</option>
6
7     {/foreach}
8   </select>
9   {foreach $node.data_map.corresponding_simulator.content.relation_list as $key => $row}
10
11     {def $url_alias_node=fetch( 'content', 'node', hash( 'node_id', $row.node_id ) )}
12     <input id="simu-{$key}" type="hidden" value="{concat('/layout/set/nolayout/',
13       $url_alias_node.url)}" />
14
15   {/foreach}
16   <div id="simu-content">
17
18 </div>

```

La partie EzPublish se termine ici et il reste maintenant le JavaScript afin d'afficher le simulateur. Pour ce faire j'ai utilisé un appel AJAX qui permet l'affichage d'une page distante sans recharger entièrement la page actuelle. Pour savoir quelle page appeler on se sert de la liste déroulante et des URL récupérée précédemment. Ensuite j'ai nettoyé un peu ce qui serait affiché pour se contenter de ce qu'on avait besoin à savoir juste le simulateur.

FIGURE 2.3 – Chargement en AJAX du simulateur de prêt

```

1 <script type="text/javascript">
2 jQuery(document).ready(function() {
3   jQuery('#loan-choice').unbind('change');
4   jQuery('#loan-choice').bind('change', function(){
5     var simu_url=jQuery('#'+jQuery(this).children(":selected").attr("id").replace(/(choice-)/, 'simu-')).val();
6     jQuery.ajax({
7       url : simu_url,
8       type : 'GET',
9       dataType : 'html',

```

```

10      async: false ,
11      success : function(code_html, statut){
12          var content_page=jQuery(code_html).find( '.content' );
13          jQuery(content_page).find( '.second_panel .breaker' ).remove();
14          jQuery(content_page).find( '.second_panel h3' ).remove();
15          jQuery(content_page).find( '.simulator-footer' ).remove();
16          jQuery(content_page).find( '.sim-share' ).remove();
17          jQuery(content_page).find( '.mention' ).remove();
18          jQuery(content_page).find( '.simulator-mentions' ).remove();
19          jQuery(content_page).find( '.simulator .liner:eq(1)' ).remove();
20          jQuery( '#simu-content' ).html( content_page.html() );
21          jQuery( '#simu-content' ).find( '.second_panel .result-line .label .required' ).attr( "style", jQuery( '#simu-content' ).find( '.second_panel .result-line .label .required' ).attr( "style" ).replace(/(10)/, '9') );
22          jQuery( '#simu-content' ).find( '.second_panel .result-line .label .optional' ).attr( "style", jQuery( '#simu-content' ).find( '.second_panel .result-line .label .optional' ).attr( "style" ).replace(/(10)/, '9') );
23          jQuery( '#simu-content' ).find( '.compatibility' ).attr( "style", jQuery( '#simu-content' ).find( '.compatibility' ).attr( "style" ).replace(/(270)/, '140') );
24          jQuery( '#simu-content' ).find( '.highlight' ).attr( "style", jQuery( '#simu-content' ).find( '.compatibility' ).attr( "style" ).replace(/(270)/, '140') );
25          eval(jQuery(code_html).find( '.content' ).find( "script" ).text() );
26          jQuery(document).find( '#layer_alert' ).remove();
27          $( ".simulator" ).css({ width: "100%", margin: "0" });
28          $( ".highlight" ).css({ position: "absolute" });
29          $( ".first_panel h2" ).css({ "text-align": "left" });
30          $( ".first_panel" ).css({ "display": "inline-block" });
31          $( ".second_panel" ).css({ "margin": "0" });
32          $( ".under_35" ).css({ padding: "6px", margin: "0px 10px 10px 10px", width: "348px" });
33          $( ".simulator .second_panel .result-line div.label" ).css({ "margin-left": "0px" });
34          $( ".simulator .second_panel .result-line div.input" ).css({ "margin-right": "5px" });
35          $( ".simulator .second_panel .result-line div.unit" ).css({ "margin-right": "0px" });
36      }
37  });
38  });
39  jQuery( '#loan-choice' ).trigger( 'change' );
40  });
41  </script>

```

Pour finir j'ai eu à adapter le style du simulateur car le client souhaitait un style différent des simulateurs habituels j'ai intégré un peu de CSS3 afin d'ajuster l'affichage. Une fois le développement terminé j'ai envoyé le lien au client pour qu'il puisse effectuer ses tests et nous faire un retour sur ce qui avait été effectué. Nous travaillons aussi avec un autre prestataire du client qui était en charge de l'intégration sur Facebook, il fallait donc aussi voir avec eux si tout était bon. Le client était satisfait et après quelques ajustement de compatibilité avec IE nous avons pu passer le simulateur en production afin de fournir l'URL final pour l'intégration d'une iframe³ sur la page Facebook. C'est à ce moment qu'un problème inattendu est arrivé car Facebook n'autorise l'intégration de lien uniquement en HTTPS or le site bfm.fr utilise HTTP, il donc fallu développer une configuration au niveau du serveur afin de réécrire les demande en HTTP vers HTTPS mais uniquement pour la page du simulateur que nous venions de développer car le client ne souhaitait pas que tout son site soit en HTTPS pour le moment. Une fois les tests effectués en interne nous avons pu transmettre la configuration à mettre en place sur les serveurs de production et tout était fonctionnel.

FIGURE 2.4 – Configuration serveur pour la réécriture des requêtes HTTPS en HTTP

```

1  bfm.conf (port 80)
2
3  RewriteEngine On
4  RewriteCond !on
5  RewriteRule ^/Simulation-Pret-Infirmiere$ https://{SERVER_NAME}/Simulation-Pret-Infirmiere (NC,R=301,L)
6  RewriteRule ^/Simulation-Pret-Militaire$ https://%{SERVER_NAME}/Simulation-Pret-Militaire (NC,R=301,L)
7
8  bfm-ssl.conf (port 443)
9
10 RewriteEngine On

```

3. Une iframe est le nom d'une balise HTML qui permet d'intégrer une page HTML au sein d'un autre document HTML


```

11 RewriteCond !.*bfm.*$
12 RewriteCond %{REQUEST_URI} !^/Simulation-Pret-Militaire
13 RewriteCond %{REQUEST_URI} !^/Simulation-Pret-Infirmiere
14 RewriteRule ^/?(.*) http://{{SERVER_NAME}}/$1 (NC,R=301,L,NS)

```

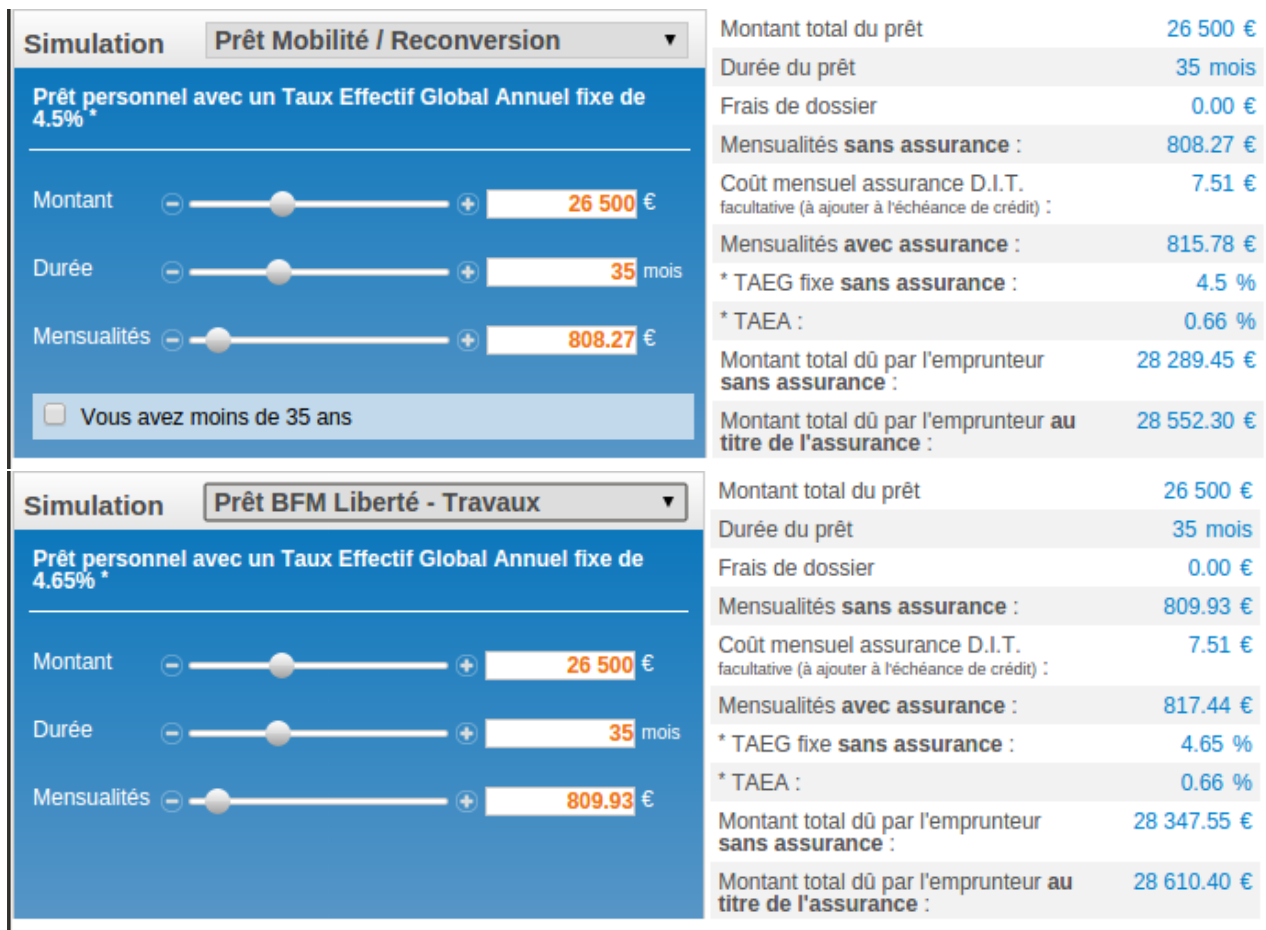


FIGURE 2.5 – Simulateur global de prêt

Convertir un simulateur d'épargne flash vers une version en HTML5/JavaScript

Pour développer ce simulateur j'ai pu reprendre du code existant et l'adapter aux besoins précis de ce cas. Le code est composé en majorité de JavaScript avec le framework jQuery afin d'afficher le simulateur et de traiter l'interaction avec l'utilisateur. Une partie de PHP propre à EzPublish est aussi présente afin de récupérer depuis le back-office les données choisies pour les taux d'épargne. Ceci étant configurable j'ai là aussi créé une classe propre à ce type de simulateur et j'ai ensuite modifié le code pour récupérer les bonnes valeurs. Il n'y a pas eu de difficultés majeurs avec cette fonctionnalité car je me basais sur de l'existant il fallait donc juste faire attention à ne récupérer que le nécessaire et ensuite la partie la plus longue était d'adapter l'affichage pour ressembler à la version originale en flash, ce que j'ai réalisé en CSS (cf. Annexe Code CSS simulateur d'épargne page 40).

BANQUE FRANÇAISE MUTUALISTE
PROFESSION BANQUIER VOCATION SOLIDARITÉ

Compte SG/BFM | Prêt personnel | Epargne | Solutions immobilières | Rechercher

Accueil > Simulateur d'épargne

Simulateur d'épargne

Vos versements

Taux de rémunération brut annuel ⁽¹⁾

1.75% Jusqu'à 3 000 €

1.25% Au delà 3 000 €

Montant du dépôt initial (minimum 1500 €) €

Montant des versements mensuels € X mois

Durée totale d'épargne mois

Calculez

(1) Taux de rémunération brut en vigueur à compter du 01/08/2014 susceptibles de variation. Intérêts soumis à l'impôt sur le revenu et aux prélèvements sociaux ou prélèvements libératoires forfaitaires.*Les résultats de cette simulation sont donnés à titre indicatif et ne sont pas contractuels.

Recevoir nos offres

BFM Profession banquier Vocation solidarité BFM et MOI Nos mutuelles sociétaires

Nous contacter | Presse | Newsletter | FAQ | Nos partenaires | Cookies | Informations légales

Des informations commerciales? Nous vous rappelons en moins de 10 minutes

Téléphone > OK

Mentions CNIL

BANQUE FRANÇAISE MUTUALISTE
PROFESSION BANQUIER VOCATION SOLIDARITÉ

Compte SG/BFM | Prêt personnel | Epargne | Solutions immobilières | Rechercher

Accueil > Simulateur d'épargne

Simulateur d'épargne

Simulateur d'épargne

Taux de rémunération brut annuel ⁽¹⁾

1.75% Jusqu'à 3 000 €

1.25% Au delà 3 000 €

Capital disponible au bout de 12 mois 2 035.00 €

Dont versements 2 000.00 €

Et intérêts bruts 35.00 €

En années

0 1

(1) Taux de rémunération brut en vigueur à compter du 01/08/2014 susceptibles de variation. Intérêts soumis à l'impôt sur le revenu et aux prélèvements sociaux ou prélèvements libératoires forfaitaires.*Les résultats de cette simulation sont donnés à titre indicatif et ne sont pas contractuels.

Imprimer | Partager par mail | Nouvelle simulation

Recevoir nos offres

BFM Profession banquier Vocation solidarité BFM et MOI Nos mutuelles sociétaires

Nous contacter | Presse | Newsletter | FAQ | Nos partenaires | Cookies | Informations légales

Des informations commerciales? Nous vous rappelons en moins de 10 minutes

Téléphone > OK

Mentions CNIL

FIGURE 2.6 – Simulateur d'épargne

Développer un template intégré à EzPublish pour la création de landing page et réaliser un exemple en s'inspirant de l'existant

Dans le cadre de la création de landing page le client avait l'habitude de développer des pages à part hors du CMS puis de les déposer sur le serveur. Ceci pose problème pour la maintenance et rend la page non contribuable depuis le back-office d'EzPublish. J'ai donc commencé par créer une nouvelle classe dans EzPublish pour les landing page et j'ai

ensuite codé un nouveau layout avec uniquement les entêtes de la page et la structure HTML. Le contenu de la page devant être entièrement contribuable depuis le back-office le code était assez simple et consistait juste à récupérer les données. Afin de vérifier le bon fonctionnement j'ai intégré les landing page existantes directement dans le CMS.



FIGURE 2.7 – Landing page

Développer un nouvel élément de formulaire permettant la redirection vers une page produit existant et transmettant les données déjà saisies par l'utilisateur

Ce développement était un des plus complexes à réaliser car presque tous les éléments devaient être configurables depuis le back-office et les données capables de passer d'un formulaire à l'autre. J'ai donc commencé par créer la classe dans EzPublish avec un bouton radio oui/non afin de lancer la redirection vers un formulaire existant. Ensuite un champ pour sélectionner le formulaire vers lequel pointer et un champ pour le message à afficher lors de la redirection. J'ai ensuite créé le type d'élément à ajouter dans les formulaires que j'ai nommé «Produit Existant». C'est ici que l'essentiel du code s'est écrit, l'idée étant de récupérer les questions du formulaire vers lequel on pointe puis de comparer si les champs existent aussi dans le formulaire actuel. Si tel est le cas alors je transmet les données saisies par l'utilisateur lors de la redirection si celle-ci est actionnée.

La première partie de récupération des champs existant sur le formulaire distant se fait grâce aux méthodes fournies par le CMS et on crée ainsi des champs cachés qui permettront la transmission des données.

FIGURE 2.8 – Récupération des questions existantes sur l'autre formulaire

```

1 {def $questions = fetch('content','list',hash(
2   'parent_node_id',$node.data_map.redirect_url.content.main_node.node_id,
3   'sort_by',$node.sort_array,
4   'class_filter_array',ezini('Classes','QuestionClasses','smileform.ini'))
5 )}
6 {def $atts = fetch('content','list',hash(
7   'parent_node_id',$node.parent_node_id,
8   'sort_by',$node.sort_array,
9   'class_filter_array',ezini('Classes','QuestionClasses','smileform.ini'))
10 )}
11
12 {foreach $atts as $att}
13   {foreach $questions as $question}
14     {if $question.name|lowercase()|begins_with($att.name|lowercase())}
15       <div class="hidden">
16         {if $att.data_map.contentclass_attribute_identifier|eq('gender')}

```

```

17         {node_view_gui content_node=$question view='form' default_value='madame'
18         }
19     { else }
20         {node_view_gui content_node=$question view='form' default_value=''}
21     { / if }
22 </div>
23 { / if }
24 { / foreach }

```

Encore une fois la deuxième partie du développement s'est effectuée en JavaScript avec le framework jQuery. Je devais récupérer les données saisies par l'utilisateur à chaque fois qu'il quittait un champ puis les transmettre dans les champs récupérés cachés du formulaire vers lequel on pointe. Ensuite on vérifie si l'utilisateur choisi l'option «Produit Existant» et si tel est le cas on affiche la popin avec le message indiquant la redirection vers le nouveau formulaire tout en transmettant les données déjà saisies (cf. Annexe Transmission de données d'un formulaire page 42).

Développer une page Web à disposition des téléconseillers intégrant une page d'orientation vers les différentes simulation de prêts personnels et une fonctionnalité d'envoi de mail type intégrant un lien vers une simulation avec des paramètres intégrés dans l'URL

Cette page se décompose en deux parties, il en sera donc de même pour le code. Dans un premier temps une section où l'on choisie le prêt que l'on souhaite puis on indique le montant et la durée de celui-ci et enfin on le simule. Ceci génère une URL avec les paramètres intégrés et ouvre une popin avec le simulateur et les valeurs choisies. Le sélecteur de prêt se génère en récupérant les données du back-office qui est configurable en choisissant les prêts à intégrer. Ensuite avec du Javascript on récupère les valeurs saisies par l'utilisateur et on concatène celles-ci à l'URL originale du simulateur pour en former une nouvelle intégrant ces paramètres.

The screenshot shows the BFM website's 'Page Simulateurs'. The header includes the BFM logo and navigation links. The main content area features a form for simulating a loan. The form has several sections:

- Choix du prêt ***: A dropdown menu set to 'Prêt personnel Auto'.
- Montant ***: A text input field with a Euro symbol (€).
- Durée ***: A text input field with 'mois' (months) as a unit.
- SIMULER**: A blue button to initiate the simulation.
- Adresse générée**: A large text input field for the generated address.
- Civilité ***: Radio buttons for 'Mme' and 'M'.
- Nom ***: A text input field.
- Prénom ***: A text input field.
- Adresse e-mail ***: A text input field.
- Choix du prêt ***: Another dropdown menu.
- URL du simulateur ***: A text input field.
- Durée ***: A text input field.
- Montant ***: A text input field.

Below the form, there is a red text label 'Champs obligatoires(*)' and a blue 'Envoyer' button. The footer contains a navigation bar with links like 'Recevoir nos offres', 'BFM', 'Profession bancaire', 'Vocation solidarité', 'BFM et MOI', 'Nos mutuelles sociétaires', and a sidebar with 'Des informations commerciales?' and 'Nous vous rappelons en moins de 10 minutes'.

FIGURE 2.9 – Pages téléconseillers formulaire

La deuxième section de cette page est un formulaire qui récupère toujours en JavaScript les données saisies pour la simulation plus haut ainsi que l'URL générée et permet de remplir les coordonnées du destinataire et de lui envoyer toutes les informations par mail. Ce mail devant être configurable dans le back-office et offrir la possibilité de récupérer les informations saisies par l'utilisateur et de les réintégrer dans le mail aussi bien pour l'utilisateur que pour le webmaster. Nous avons rencontré des problèmes de comptabilité avec IE pour l'ouverture de la popin intégrant le simulateur, en effet le code initial était compatible avec Chrome et Firefox mais pas avec IE9 il a donc fallu le repenser afin de le rendre fonctionnelle sous tous les navigateurs (cf. Annexe Illustrations pages téléconseillers page 43).

Bilan du projet

Pour ce second travail avec EzPublish j'ai eu l'occasion d'exercer sur un projet d'envergure et de longue haleine. En effet BFM étant un client de TMA de nombreuses corrections, améliorations ou évolutions sont sans cesse à apporter. Ceci offre une vraie diversité dans le projet et le rend donc très intéressant en soi mais aussi pour se former sur la technologie car il permet de couvrir une grande partie des possibilités du CMS.

D'un point de vue relationnel ce projet m'a permis d'aborder un nouvel aspect du métier d'ingénieur, à savoir la relation avec le client. J'ai été amené à plusieurs reprises à avoir des rendez-vous téléphoniques afin de discuter de l'avancement des tâches, des modifications

à effectuer ou même parfois d'expliquer certains points techniques au client pour qu'il puisse configurer le site à ses souhaits.

Pour ce qui est du technique, je suis rentré véritablement dans le coeur du métier avec du PHP, du JavaScript et du style. Autant de langages peu vu en cours mais relativement simple à prendre en main que ce soit en s'inspirant de l'existant ou en recherchant sur internet. Le CMS lui aussi m'était inconnu mais j'ai pu compter sur les autres membres du projet pour m'aider et me conseiller ce qui m'a permis d'appréhender les tâches à effectuer dans de bonnes conditions. J'ai donc beaucoup appris sur ce projet que ce soit sur le plan humain ou technique et même si par la suite je vais être amené à changer de technologie et passer sur du e-commerce avec Magento, il n'en reste pas moins que les langages du WEB sont les mêmes notamment le JavaScript, CSS ou HTML et que PHP est aussi le langage serveur utilisé.

Je noterai quand même quelques difficultés rencontrées au cours de ce projet, par exemple les appels au client qui n'étaient pas simple au début par manque d'expérience dans ce domaine. Il est difficile de se mettre à la portée de la personne que nous avons au téléphone et d'être clair, précis et concis quand nous expliquons le problème rencontré ou les actions à effectuer. Enfin la difficulté technique majeure de ce projet et qui à fortiori à engendrer une complication de communication, est que nous n'avions pas accès aux serveurs de productions et de pré-production comme je l'ai expliqué plus tôt. Ceci a rendu difficile les processus de livraison où le client lui même devait effectuer certaines manipulations et nous a obligé à être très précis et patient dans nos explications. Parfois nous avions des bugs que nous n'arrivions pas à reproduire en interne et ne pouvions pas voir ce qu'il se passait dans les logs⁴ rendant la tâche de debug⁵ compliquée. Bilan positif encore une fois et qui me motive pour le prochain projet qui m'attend avec en plus la découverte d'une nouvelle technologie, Magento.

4. Le concept de log est l'enregistrement dans un fichier précis de certaines données ou d'événements que le développeur a besoin de connaître, sans pour autant affecter ou s'afficher sur le programme principal. Dans notre cas de site Web ce sont des données qui ne s'afficheront pas en front par exemple

5. Action d'analyse et de correction des bugs

2.3 POC (Proof Of Concept) thème responsive Gifi

Présentation du projet et état de l'art

L'idée de ce « mini-projet » ou POC était de vérifier la faisabilité d'un changement de thème sans modifier la version de l'application. Le site e-commerce de Gifi étant dans une ancienne version de Magento 1.13 sans thème responsive d'intégré. Nous voulions récupérer le thème responsive natif de la nouvelle version de Magento 1.14 et vérifier que sa mise en place ne casse pas le design et le fonctionnement actuel du site.

Mes objectifs

- Intégrer le thème responsive d'un Magento 1.14 sur un Magento 1.13

Mes réalisations

J'ai donc commencé par récupérer la version correspondante à celle mise en place sur le site de Gifi (Magento 1.13), je l'ai installé sur ma LXC afin de pouvoir commencer le test. Ensuite j'ai copié le thème responsive de Magento 1.14 et je l'ai chargé depuis le back-office. Il s'agissait ensuite d'effectuer des tests de compatibilité pour voir si des modules natifs de Magento ne fonctionnait plus. Les fiches produits notamment ne s'affichait plus correctement car une fonction n'était plus déclaré au même endroit dans le code. J'ai donc remonté ces petits bugs et confirmé la faisabilité de ce « backportage » de thème. J'en ai profité pour me former sur Magento en prévision de la formation et des projets à venir. Sachant que j'allais être amené à travailler sur le design et le templating de sites je me devais de bien assimiler la structure des pages de cette technologie. Comme on peut le voir sur le schéma ci-dessous une page est composée d'un layout qui regroupe les différents blocs et à l'intérieur de ces blocs nous avons des templates qui gèrent l'affichage. Dans ces templates on peut aussi trouver des helpers qui sont des méthodes (pouvant être issues de d'autres blocs) retournant des données.

Layout

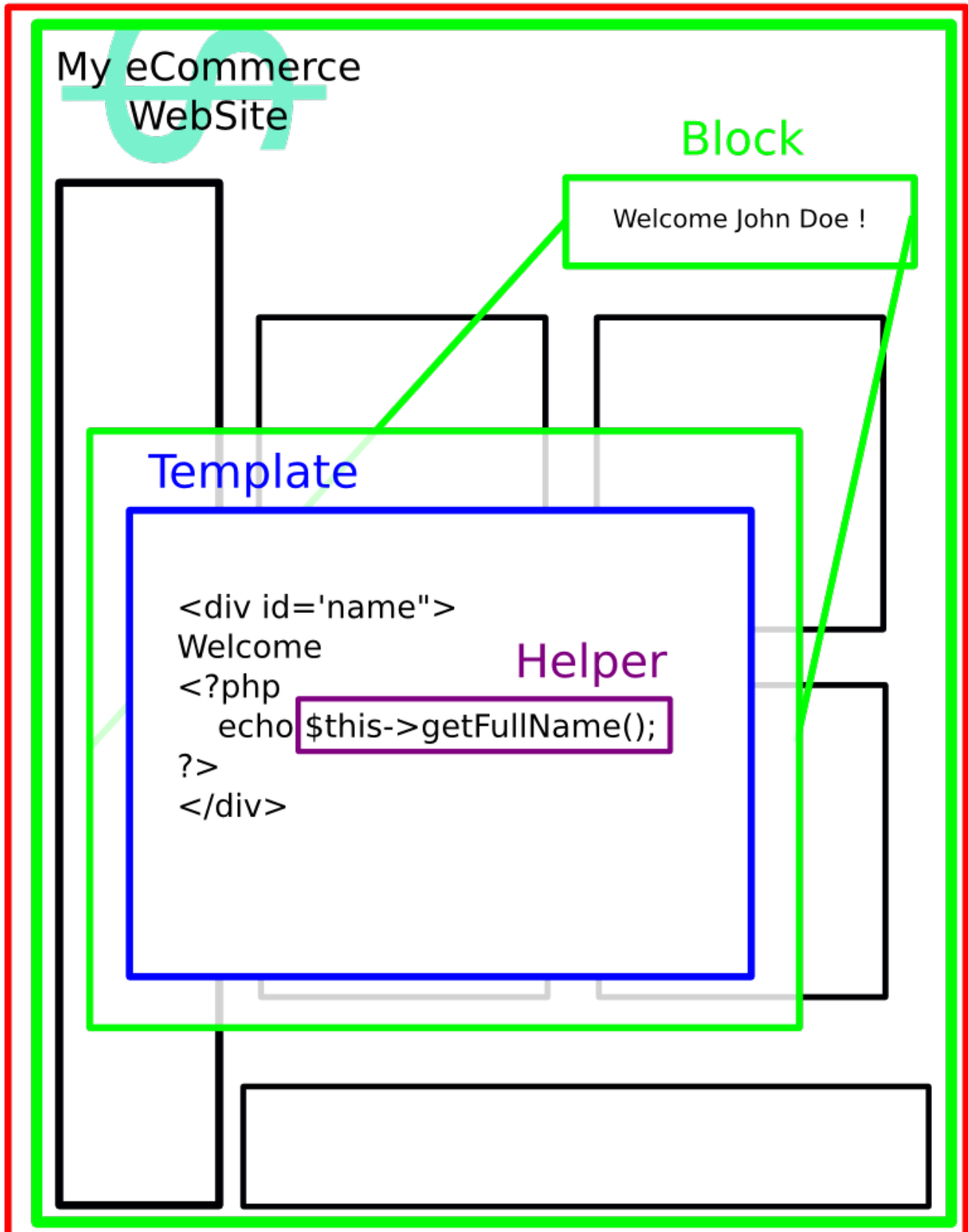


FIGURE 2.10 – Structure de base d'une page Magento

Bilan du projet

Plus qu'un projet en tant que tel j'ai découvert par le biais de cette mission une autre facette du métier. En effet il n'est pas rare que les clients demandent ce genre d'expertise afin de

savoir approximativement la faisabilité d'une évolution et surtout son coût. Car le POC a aussi pour vocation de donner un ordre d'idée du temps nécessaire à la réalisation et donc le budget à y consacrer. C'est donc un bon moyen pour le client de se rendre compte si l'investissement en vaut la peine ou s'il est préférable de se passer de ce développement et de consacrer le budget à autre chose. D'un point de vu purement technique cela m'a permis de prendre en main Magento et de bien assimiler les notions de thème, de layout et de template qui sont des aspects clés des projets où je serai amené à travailler par la suite.

Magento This is a demo store. Any orders placed through this store will not be honored or fulfilled.

YOUR LANGUAGE: English WELCOME

ACCOUNT CART

Search entire store here...

WOMEN MEN ACCESSORIES HOME & DECOR SALE VIP

HOME / MEN / SHIRTS

SHOP BY

PRICE

140,00 € - 149,99 € (1)

160,00 € - 169,99 € (1)

190,00 € and above (1)

COLOR

(1) (1)

(1) (1)

(1) (1)

OCCASION

Career (1)

Casual (1)

Evening (1)

TYPE

Shirts (3)

SLEEVE LENGTH

Long Sleeve (3)

SHIRTS

SORT BY: Position VIEW AS: 3 Item(s) SHOW: 12

PLAID COTTON SHIRT 160,00 €

VIEW DETAILS

Add to Wishlist

Add to Compare

SLIM FIT DOBBY OXFORD SHIRT 175,00 € 140,00 €

VIEW DETAILS

Add to Wishlist

Add to Compare

FRENCH CUFF COTTON TWILL OXFORD 190,00 €

VIEW DETAILS

Add to Wishlist

Add to Compare

SORT BY: Position VIEW AS: 3 Item(s) SHOW: 12

FIGURE 2.11 – Theme responsive Magento 1.14 demo

2.4 Refonte graphique Darjeeling

Présentation du projet et état de l'art

Darjeeling fait partie des clients de Smile depuis relativement longtemps, le site initial a été réalisé par le groupe. Il s'agit d'un site e-commerce de lingerie proposant de nombreux produits et les services associés, il est à noter que des magasins physiques sont présents partout en France. Aujourd'hui nous intervenons dans le cadre d'une refonte graphique et allons modifier principalement les templates, les styles et s'assurer de ne pas dégrader le résultat sur mobile. Il n'y a donc pas ou très peu de développement fonctionnel sur ce projet, uniquement du visuel.

Les technologies utilisées n'en sont pas moins intéressantes, notamment le système mis en place pour la compilation des fichiers de style et JavaScript. En effet nous utilisons Sass (Syntactically Awesome Style Sheets) qui est une sorte d'amélioration du langage CSS et qui y ajoute certaines fonctionnalités. Ces fichiers doivent être ensuite compilé pour obtenir un CSS qui est le seul langage compris par les navigateurs. Pour ce faire nous utilisons Grunt qui surveille toutes modifications apportées aux fichiers Sass et qui non seulement les compile mais les minifie aussi. Au final nous obtenons un seul et unique gros fichier CSS qui est la concaténation de tous les fichiers Sass. Ceci permet de développer sur des fichiers séparés et clairs puis de livrer au serveur un fichier unique ce qui est préférable en terme de performance. Comme la plupart des projets la principale difficulté sera le temps, car nous avons eu un peu plus de 2 mois pour réaliser le projet.

Mes objectifs

- Refonte du template et du style de divers pages du site
- Refonte de la liste produits et de ses filtres
- Optimisation du nouveau design sur mobile
- Phase de recette avec le client

Mes réalisations

Refonte du template et du style de divers pages du site

Il s'agit pour moi de mon premier gros projet en utilisant Magento, j'ai donc commencé par réaliser la refonte des pages les plus simples à savoir le header et le footer. J'ai donc redessiné ces pages en suivant les maquettes graphiques fournies par le client. Il y avait notamment des changements de logos, d'icônes, de couleurs et de polices mais aussi quelques modifications fonctionnelles comme le changement des liens présents dans le header et le menu de navigation (cf. Annexe Illustration footer Darjeeling page 45). J'ai eu par exemple à ajouter des champs de le back-office permettant au contributeur d'ajouter une image et un lien de promotion directement dans le menu. Ceci doit être configurable, j'ai donc commencé par développer un « installer » qui permet l'ajout d'attributs dans le back-office (par modification de la base de données), un pour le choix d'une image et l'autre pour le lien vers lequel pointe cette image (cf. Annexe Installateur Magento pour ajouter un attribut au BO⁶ page 46). J'ai ensuite modifié le template du menu afin de faire appel à ces attributs et ainsi afficher le contenu ajouté. Finalement j'ai adapté le style pour respecter la mise en forme souhaitée par le client.

6. Back-Office



FIGURE 2.12 – Nouveau menu de navigation Darjeeling

Pour le footer la partie mise en forme était assez simple, du style CSS sans complexité, la fonctionnalité intéressante réside dans la récupération des liens que l'on affiche dans celui-ci. Un attribut position avait été ajouté et je l'ai donc utilisé afin d'offrir la possibilité au contributeur de choisir dans quelle partie du footer il souhaite positionner le lien. Il ne me restait plus qu'à mettre en place une boucle dans le template qui parcourt les éléments et récupère leur lien pour les afficher.

FIGURE 2.13 – Récupération liens à afficher dans le footer depuis le back-office

```

1 <?php
2 $_pagesDarjeeling = $this->getPagesDarjeeling(); // Recuperation from BO
3 ?>
4
5 <ul>
6     <li class="category">
7         <a href="<?php echo $this->getUrl('conseils'); ?>"><?php echo $this->__( 'Our tips ' ); ?></a>
8     </li>
9     <?php if ( $_pagesDarjeeling && $_pagesCountDarjeeling > 0 ) : ?> // Verification of existing
        elements
10         <?php foreach ( $_pagesDarjeeling as $_pageDarjeeling ) : ;?> // Loop on elements
11             <li class="category">
12                 <a href="<?php echo $_helper->getPageUrl($_pageDarjeeling); ?>"><?php echo
                    $_pageDarjeeling->getTitle(); ?></a> // Display the link
13             </li>
14         <?php endforeach; ?>
15     <?php endif; ?>
16     <li class="category">
17         <a href="<?php echo $this->getUrl('affiliate'); ?>"><?php echo $this->__( 'Affiliation ' ); ?><
            /a>
18     </li>
19 </ul>

```

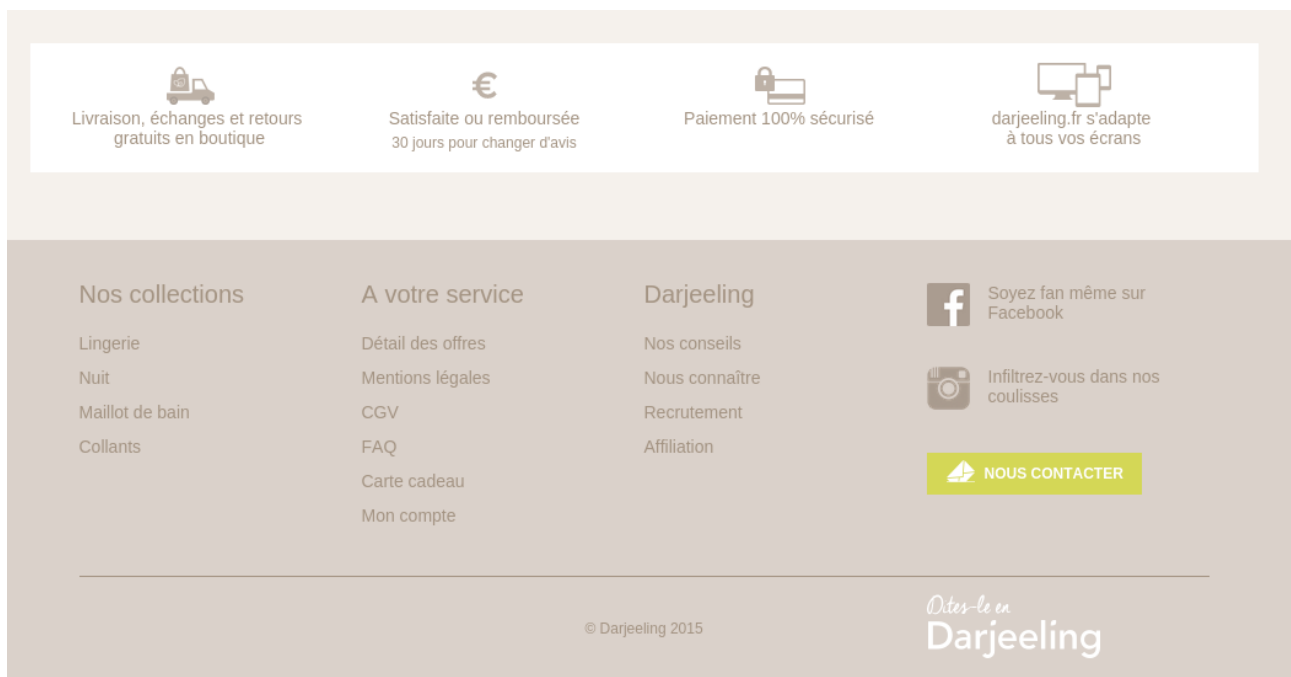


FIGURE 2.14 – Footer Darjeeling

Par la suite j'ai réalisé la page « Mes envies » qui regroupe les produits sélectionnés par l'utilisateur et lui permet soit de les ajouter à son panier soit de les retirer de cette liste. J'ai donc redesigné la page ainsi que les blocs qui affiche les produits et implémenté un script pour l'affichage d'info bulle custom. Il s'agit uniquement de développement front, du templating et du style mais aussi du JavaScript. Sachant que cette info bulle devrait être présente à plusieurs endroits dans le site j'ai mis en place un petit script jQuery permettant de vérifier si le produit est déjà dans la liste « Mes envies » au quel cas on propose d'enlever le produit. Dans le cas contraire l'info bulle affiche « Ajouter à mes envies ». Ce script doit être capable de réagir au clique sur le bouton et au chargement de la page, j'utilise pour cela les observateurs jQuery (cf. Annexe Script jQuery d'ajustement d'une info bulle page 47).

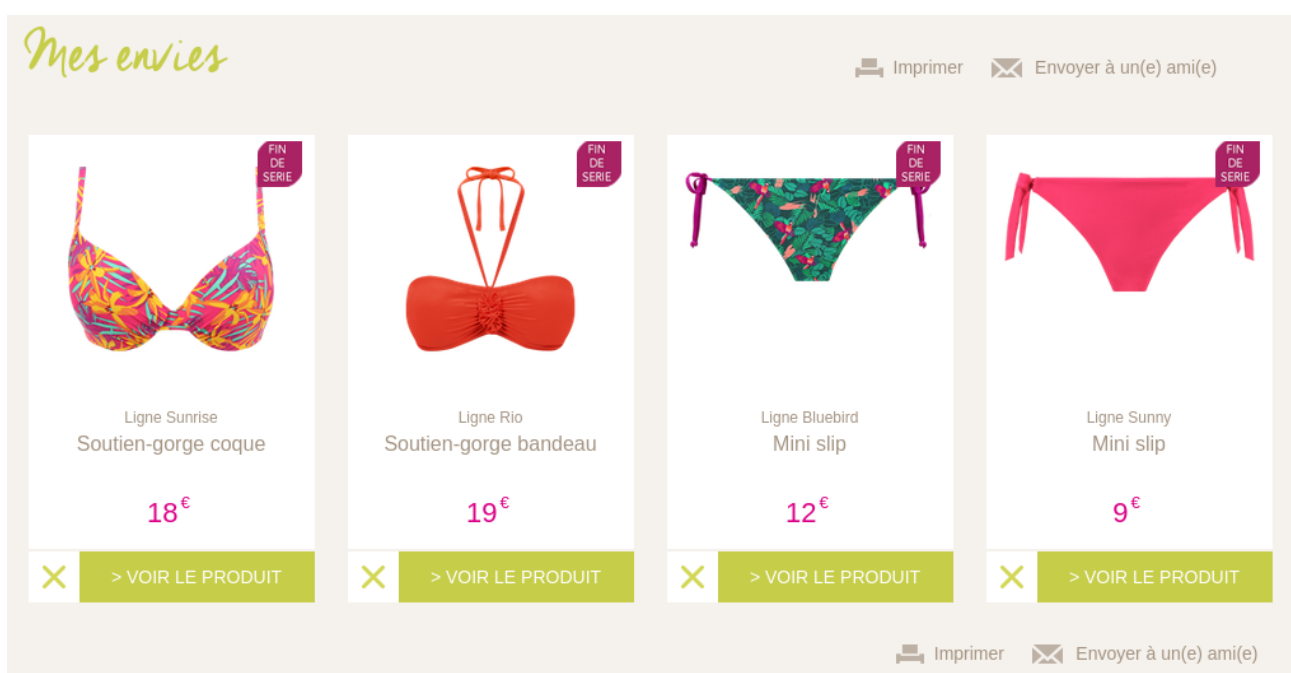


FIGURE 2.15 – Page « Mes envies » Darjeeling

Refonte de la liste produits et de ses filtres

La liste produits est un template appelé à différents endroits du site notamment lorsque l'on sélectionne une catégorie particulière ou que l'on effectue une recherche. Ce template est chargé de lister les produits comme son nom l'indique, de mettre en place une pagination et de proposer divers filtres pour trier les résultats. Pour ce qui est de l'affichage j'ai mis en place un nouveau design de blocs sur 3 colonnes respectant les maquettes. Le client souhaitait afficher 48 produits par pages et proposer un lien en bas permettant de charger les 48 produits suivants. J'ai donc implémenté cette fonctionnalité par le biais d'un appel AJAX qui charge les éléments sans avoir à recharger toute la page. J'ai aussi adapté le template pour répondre à ces nouveaux besoins (cf. Annexe Illustration voir plus de produits Darjeeling page 48).

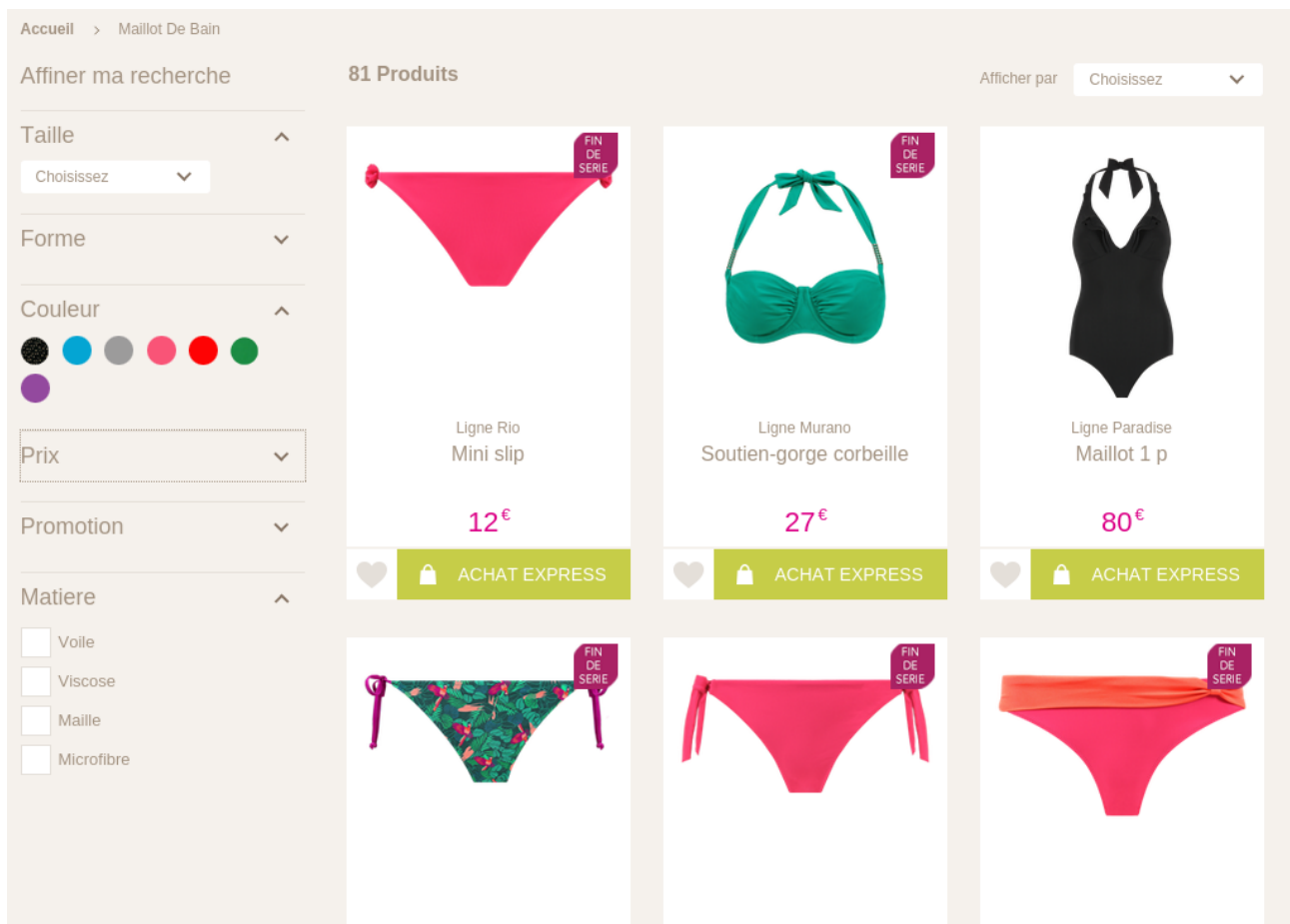


FIGURE 2.16 – Liste produits

Les blocs affichant les produits ont aussi été modifiés en ajoutant de nouvelles fonctionnalités d'achat et de déplacement dans la liste de souhait. J'ai notamment utilisé le script développé et cité précédemment. Concernant la partie filtre j'ai repris le fonctionnement qui était déjà en place en terme de fonctionnalités et j'ai ajouté des éléments de design en plus de la refonte graphique. Ceci s'est traduit par l'utilisation de JavaScript pour mettre en place une liste déroulante permettant d'afficher ou non le détail des filtres. Ainsi qu'un script qui regarde qu'elles filtres sont actifs au changement ou rechargement de la page et les réouvre directement dans le but améliorer l'expérience utilisateur.

FIGURE 2.17 – Script jQuery d'affichage en accordéon des filtres

```
1 <script>
2 jQuery( window ).unload(function () {
3     var i = 0;
4     var numberActive = {};
5     jQuery.each(jQuery( '#multiAccordion h3' ),function(){
```

```

6         if ( jQuery(this).next('div').css('display') == 'block' ) {
7             numberActive.push(i);
8         }
9         i += 1;
10    });
11    sessionStorage.setItem('accordion-active-items', numberActive);
12 });
13 jQuery( document ).ready(function () {
14     var numberActive = sessionStorage.getItem('accordion-active-items');
15     var defaultNumberActive = ("0,");
16     var activeItems = numberActive.concat(defaultNumberActive);
17     jQuery('#multiAccordion').multiAccordion({
18         active: activeItems,
19         click: function(event, ui) {
20             },
21         init: function(event, ui) {
22             },
23         tabShown: function(event, ui) {
24             },
25         tabHidden: function(event, ui) {
26             }
27     });
28     jQuery('#multiAccordion').multiAccordion("option", "active", activeItems);
29     sessionStorage.clear();
30 });
31 </script>

```

Optimisation du nouveau design sur mobile

Le site mobile de Darjeeling venait tout juste d'être refait par Smile, il s'agissait donc juste d'impacter les principaux changements graphiques sur cette plateforme. Uniquement du développement graphique, changement de couleurs, de logos,... Enfin j'ai dû m'assurer que les changements fonctionnels sur le site desktop ne cassait pas le fonctionnement sur mobile puisque certains modules sont repris dans les deux cas.

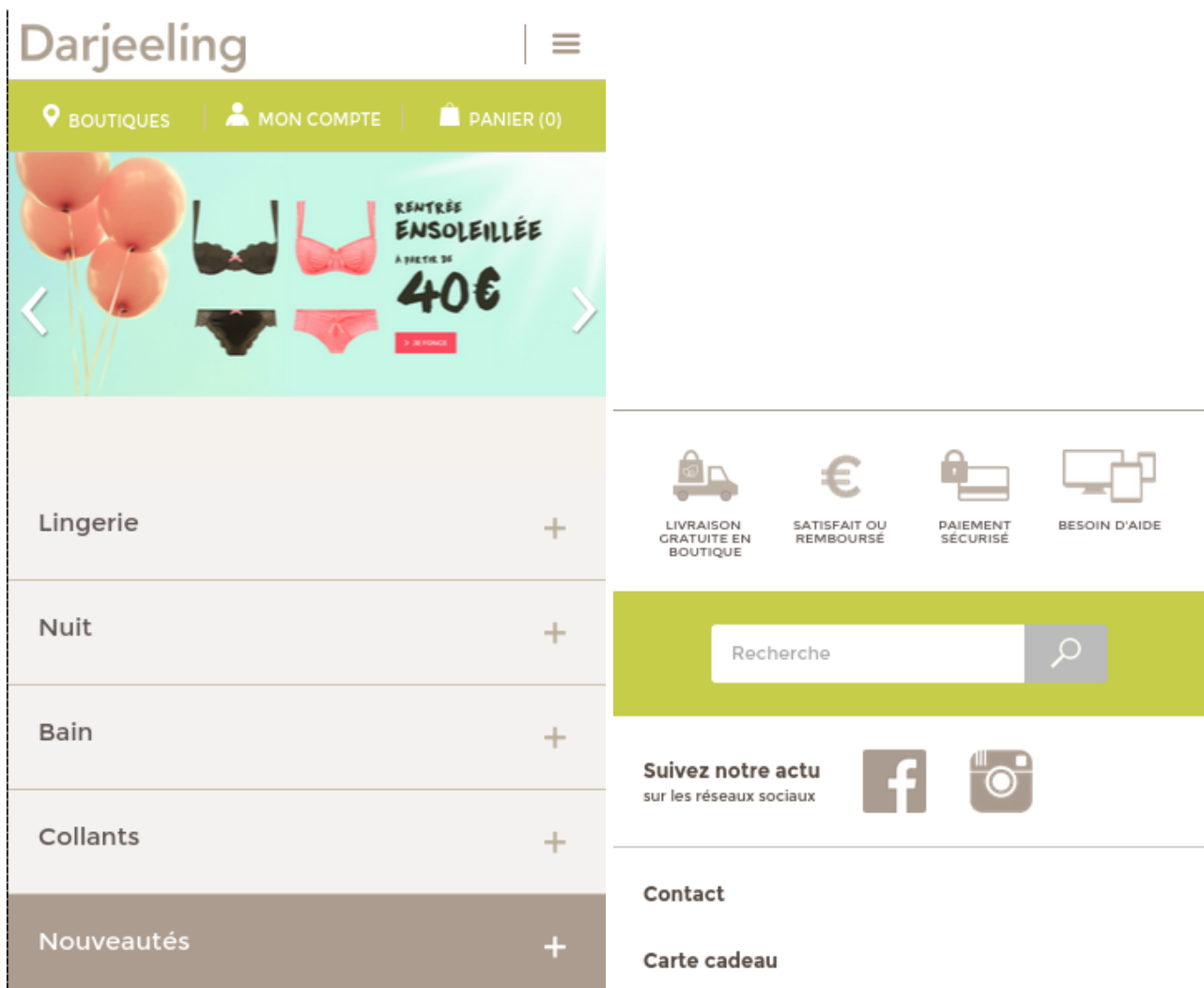


FIGURE 2.18 – Adaptation du nouveau design sur mobile

Phase de recette avec le client

Une fois les développements terminés nous avons passé le projet sur le serveur de préproduction afin que le client puisse regarder et tester ce qui avait été fait. L'intérêt de cette phase est d'avoir les retours du client et si une anomalie est détectée il crée un ticket sur Redmine (notre plateforme de gestion de projet) nous indiquant le problème rencontré. Celui-ci est ensuite vérifié par le chef de projet pour s'assurer qu'il s'agit bien d'une anomalie rentrant dans le périmètre du projet et non une évolution à facturer en plus au client. Dans le premier cas le ticket est affecté à un développeur qui va corriger le problème puis le relivrer pour que le client puisse s'assurer que tout est réglé. L'intérêt pour moi durant cette phase du projet a été de pouvoir évoluer sur l'ensemble du site et non uniquement sur le développement que j'avais effectué lors de la phase précédente. En effet j'ai été amené à corriger des bugs, apporter des modifications ou même ajouter de nouvelles fonctionnalités sur des pages qui ne me concernaient pas initialement.

Par exemple l'ajout d'un lien configurable sur un widget de la home page. Je n'avais pas travaillé avec les widget tout au long du projet et là, ça a été l'occasion pour moi de les découvrir. J'ai donc ajouté un attribut au widget permettant au contributeur de sélectionner si oui ou non il souhaite que le lien s'affiche. Puis dans le template ajouter une condition sur l'affichage du lien en récupérant l'information saisie dans le BO.

Bilan du projet

La mise en production du site à été une réussite, on peut donc dire que le projet s'est bien déroulé car l'objectif final est atteint et le client est satisfait. La période de garantie et la TMA seront là pour corriger les éventuels problèmes qui pourraient survenir. D'un point de vu personnel je suis satisfait de mes réalisations sur le projet étant donné les délais, notamment sur la phase de recette où j'ai du faire preuve de réactivité face aux retours du client. Je me suis rendu compte par le biais d'échanges avec le chef de projet qu'il était compliqué de gérer les problèmes relevés par le client. En effet il faut savoir dire non, lorsqu'un ticket relève d'une évolution et non d'une correction. La connaissance du périmètre du projet est essentielle et sa frontière parfois mince. Le temps et le budget étant limité nous ne pouvons pas tout faire sans dépasser du planning même si parfois en tant que développeur nous aimerions optimiser le rendu.

Ce projet m'a énormément enseigné sur Magento grâce à la collaboration avec les autres membres du projet, ce qui m'a permis de progresser rapidement et de devenir autonome sur la plupart des tâches. J'ai ainsi pu réaliser la recette par moi même pour ce qui est des problèmes liés au design du site. J'ai eu l'occasion d'aborder certains points fonctionnels de Magento ce qui me sera utile pour la suite de mon stage et surtout j'ai acquis de méthodes de travail pour le debug. De nombreux moyens sont mis à dispositions par Magento pour détecter et corriger les problèmes (notamment les logs) et j'ai appris à m'en servir efficacement. Aussi, ma compréhension de l'architecture et du fonctionnement global de cette technologie e-commerce s'est éclaircie, ainsi que le enjeux métiers relatif au commerce sur internet.

Les principaux problèmes rencontrés sont liés aux anciens développements du site, qui a été construit de manière peu conventionnel, rendant parfois notre tâche plus compliquée qu'elle ne semblait l'être. Cependant j'ai pu compter sur mes collaborateurs qui avaient de l'expérience sur le sujet pour m'aider. La seconde difficulté était le planning à tenir, celui-ci était très serré et nous a parfois obligé à faire des concessions sur la qualité du code ou sur les demandes du client. Enfin le dernier point compliqué à gérer pour moi était de mener deux différents projets en même temps car en attendant les retours de la part du client pour la recette j'ai entamé le nouveau projet sur lequel j'ai été affecté (que je détaillerai par la suite). Il a donc fallu jongler entre ces deux plateformes de e-commerce sans se mélanger ce qui était une première pour moi.

2.5 Snowleader refonte 2015

Présentation du projet et état de l'art

Snowleader, c'est la référence en e-commerce pour le snow, la glisse et l'outdoor avec un catalogue de plus de 15000 références. Pour Smile Sud-Ouest cela est synonyme d'un gros projet avec de nombreuses demandes et aussi bien sur le design que sur les fonctionnalités. Comme sur les autres projets nous avons des contraintes de temps bien entendu mais aussi de performance car le catalogue est vaste ce qui fait beaucoup de données à traiter et enfin de compatibilité, en effet le design original doit s'afficher correctement sur tous les navigateurs et être responsive sur tablette (nous ne nous occupons pas du mobile car un site à part y est dédié).

Qui dit projet conséquent, dit de nombreuses attentes de la part du client, de la précision en terme de rendu et donc une charge de travail élevée pour l'agence. C'est donc avec un équipe projet conséquente que le projet a été mené soit une dizaine de personnes, ce qui représente presque un quart des collaborateurs de Smile Sud-Ouest. Pour ma part je n'ai pas rejoint le projet dès son commencement car je travaillais sur autre chose, je suis venu en renfort au vu des besoins de développement. En effet la majorité des pages du site sont à refaire aussi bien en terme de design que d'agencement ou de fonctionnalité. Nous avons à notre disposition bon nombre de maquettes graphiques ainsi que des spécifications techniques détaillées pour nous aider dans notre travail.

Ma tâche principale est l'espace client, ce qui regroupe les pages de connexions et de déconnexions, le dashboard avec l'accès à toutes les données clients (historique de commandes, points de fidélités, ...) et les divers formulaires d'inscriptions ou de modification d'informations.

Mes objectifs

- Refonte de l'historique des commandes et de leurs détails
- Création d'une nouvelle page « Fidélités » regroupant plusieurs blocs
- Intégration conditionnelle de blocs CMS
- Gestion des informations clients

Mes réalisations

Refonte de l'historique des commandes et de leurs détails

Nativement Magento propose aux clients un espace qui leur est réservé, un compte doit être préalablement créé par l'utilisateur pour pouvoir y accéder. Dans cet espace, il y retrouve notamment l'historique de ses commandes, passées ou en cours. Le client avait déjà redesigné cet espace lors de la version précédente, nous sommes donc loin de l'aspect natif. Cependant de nouvelles améliorations sont à apporter, en terme d'aspect, ce qui va être réalisé en suivant les maquettes graphique que nous fournient en prestataire du client. En terme de données à afficher aussi, car la nouvelle interface se veut plus épurée que l'existante (cf. Annexe Illustration dashboard Snowleader page 49).

En terme de complexité, une des partie délicate a été le bouton de gauche (pentagone avec une croix à l'intérieur) que j'ai réalisé uniquement en CSS pour éviter de mettre une image et ainsi gagner en performance. Il faut en effet faire attention car le design original proposé par l'agence graphique est certes très jolie mais peut vite devenir problématique en terme de performance et d'implémentation. Pour le reste de ce tableau j'ai repris l'existant et joué avec le style car les données affichées sont restées les mêmes. La seule chose que j'ai ajouter c'est un lien vers une page d'impression, qui vérifie en PHP si le commande

possède une facture, auquel cas le lien redirige vers la page d'impression de la facture et dans le cas contraire vers la page d'impression de la commande (cf. Annexe Condition sur le lien d'impression de facture page 50).




FIGURE 2.19 – Page d'historique des commandes

Ensuite nous rentrons dans le détail d'une commande précise, il s'agit d'une autre page, là aussi native de Magento. A l'intérieur de celle-ci nous trouvons les informations sur le client, son adresse de livraison et de facturation, les produits qui ont été commandé avec leur prix, leur quantité, ... La partie, produits commandés est restée relativement semblable au natif mise à part le style qui a été complètement revu. Là encore l'utilisation d'un tableau est le plus simple car il permet d'afficher clairement les données relatives à chaque produit. Je n'ai donc pas rencontré de difficulté majeure sur ce développement.

Sur cette même page j'ai repris la méthode développée et citée ci-dessus pour ajouter le lien vers la page d'impression avec les mêmes conditions. J'ai aussi ajouté un lien vers le suivi de la commande dans le cas où la commande a été expédiée. Pour ce faire j'utilise une méthode qui me permet de savoir si oui ou non la commande est expédiée et si tel est le cas alors je récupère son numéro de suivi que je concatène avec l'URL de suivi de colis et je l'affiche.


Grâce au BO de Magento j'ai pu tester mes différents développements, car je peux simuler depuis celui-ci le passage d'une commande puis faire avancer les étapes du processus de préparation et de livraison. Ceci m'a permis de vérifier que les diverses conditions d'affichage étaient fonctionnelles sans avoir à réellement passer de commande, ce qui me rend indépendant du reste des développements par exemple le module de paiement, le tunnel de vente, qui peuvent, eux, ne pas être fonctionnelles à l'heure actuelle (cf. Annexe Illustration back-office client Magento page 51).



COMMANDE DU : VENDREDI 24 JUILLET 2015
 Ref : 100233859

2 articles pour : **289,59 €**



COMMANDE EN PRÉPARATION



ADRESSE DE LIVRAISON
 TNT Relais Colis Express en 24h - Relais TNT - PEINTURE RODIER COLOR
 (41 RUE RODIER, 75009 - PARIS 09 - [Horaires](#))
 test test
 75000 - test
 France

ADRESSE DE FACTURATION
 test test
 75000 - test
 France

SUIVEZ VOTRE COMMANDE

PRODUIT	RÉF	PRIX UNITAIRE	QUANTITÉ	PRIX TOTAL
 SHREDITOR 92 tailleski 2-170 cm	K2_00204 2-170 cm	274,11 €	1 1	274,11 €
 MAGAZINE OFFERT	SNOW00081	0,00 €	1 1	0,00 €
Sous-total				274,11 €
Frais de port				10,00 €
Discount (Smile 15%)				-49,34 €
TVA				54,82 €
TOTAL FINAL				289,59 €

« RETOUR À MES COMMANDES »

FIGURE 2.20 – Détail d'une commande

Création d'une nouvelle page « Fidélité » regroupant plusieurs blocs

Dans un deuxième temps j'ai eu à créer une nouvelle page qui regroupe ce qui se faisait en trois pages distinctes auparavant, à savoir, une pour les points de fidélités, une pour les crédits magasins et une pour les chèques cadeaux. Dans un soucis de facilité et d'expérience utilisateur tout rassembler en un même endroit est beaucoup plus simple et clair, cette page offre en effet la possibilité de voir ses points de fidélités et leurs historiques, de voir les crédits magasins et d'accéder à la page liée comprenant l'historique là aussi et enfin d'ajouter un chèque cadeau. On trouve en plus sur cette page un bloc CMS mais j'en reparlerai dans la partie suivante.



FIGURE 2.21 – Nouvelle page Fidélité

Je ne vais pas détaillé la création de chacune des parties de peur d'être redondant mais juste expliquer la trame générale du développement et prendre l'exemple de la partie points de fidélités. J'ai donc commencé par coder un controller capable d'intercepter l'URL customer/loyalty comme demandé dans les spécifications techniques. Pour ce faire je crée un nouveau fichier PHP dans lequel je déclare ma classe, ensuite dans la configuration du module j'ajoute un nouvel élément pour « attraper » l'URL et utilisé la bonne classe. A cela il faut ajouter la création d'un layout custom dans lequel je déclare les blocs dont j'aurai besoin et préciser ce layout dans la configuration du module. Une fois cela terminé je me retrouve avec ma nouvelle page, mes différents blocs d'affichés provenant de l'existant, il ne me reste donc plus qu'à les modifier pour leur donner l'aspect et les fonctionnalités souhaitées (cf. Annexe Déclaration du controller custom loyalty page 52).

Prenons l'exemple du bloc responsable des points de fidélité, dans la classe PHP de ce bloc en question je retrouve des méthodes me permettant d'effectuer diverses actions (je peux bien évidemment ajouter les miennes). A cela il faut ajouter un fichier de template pour ce bloc qui est lui responsable de l'affichage et c'est ici que nous allons retrouver la plupart du HTML ainsi qu'un peu de PHP notamment pour réaliser des boucles ou appeler les méthodes du bloc. C'est ensuite assez simple, je récupère les données dont j'ai besoin, le nombre de points de fidélités que possède le client puis je l'affiche. Cela aurait pu être aussi simple que cela mais le client avait une demande de design un peu particulière, avec une case pour chacun des chiffres du nombre. Ceci m'a obligé à développer une méthode qui sépare le nombre en chiffres puis si c'est un nombre à trois chiffres, tous les affichés, s'il n'en possède que deux ajouter un 0 avant le premier chiffre et ainsi de suite car le design comprenait trois cases.

J'ai donc créé de A à Z une page custom en allant chercher des blocs existants puis modifier le template et le style de chacun. Ceci m'a permis de voir que le potentiel de Magento est vraiment grand et que l'on peut le modifier dans le moindre détail à condition d'y mettre le temps. J'ai pu compter sur l'aide de mes collaborateurs pour m'aiguiller sur les étapes de la création et ainsi acquérir les méthodes d'experts de la technologie.

Intégration conditionnelle de blocs CMS

Une des parties du dashboard de l'espace client est composé d'une grille de blocs CMS et j'ai donc eu à la implémenter. Rien de très compliqué en soit puisqu'il m'a suffit de les contribuer dans le BO (dans un soucis de simplicité nous avons juste mis une image mais le client peut y intégrer ce qu'il veut notamment du code HTML). Puis de les déclarer dans le layout du dashboard en enfin de les appeler depuis le template. Une méthode native de Magento permet de faire cela juste en utilisant l'identifiant du bloc CMS créé dans le BO.

FIGURE 2.22 – Déclaration d'un layout avec l'appel à un bloc CMS

```

1 <?xml version="1.0"?>
2 <layout version="0.1.0">
3     <customer_loyalty_index>
4         <update handle="customer_account" />
5         <reference name="content">
6             <block type="core/template" name="customerloyalty.wrapper" template="customerloyalty/
7                 wrapper.phtml" translate="label">
8                 <block type="enterprise_customerbalance/account_balance" name="balance" template="
9                     customerbalance/account/balance.phtml" before="-" />
10                <block type="enterprise_giftcardaccount/account_redeem" name="redeem" template="
11                    giftcardaccount/account/redeem.phtml" />
12                <block type="enterprise_reward/customer_reward" name="customer.reward" template="
13                    reward/customer/reward.phtml">
14                    <block type="enterprise_reward/customer_reward_info" name="customer.reward.info"
15                        template="reward/customer/reward/info.phtml" />
16                    <block type="snowleader_reward/customer_reward_premium" name="customer.reward.
17                        premium" template="reward/customer/reward/premium.phtml">
18                        <block type="cms/block" name="customer.reward.premium.
19                            customer_become_premium" as="customer_become_premium">
20                            <action method="setBlockId">
21                                <block_id>customer_become_premium</block_id>
22                            </action>
23                        </block>
24                    </block>
25                <block type="enterprise_reward/customer_reward_history" name="customer.reward.
26                    history" template="reward/customer/reward/history.phtml" />
27                <block type="enterprise_reward/customer_reward_subscription" name="customer.
28                    reward.subscription" template="reward/customer/reward/subscription.phtml" />
29            </block>
30        </reference>
31    </customer_loyalty_index>
32</layout>

```

La seule complexité à laquelle j'ai du faire face est que le client souhaitait faire apparaître un bloc différent dans deux cas, quand le client est premium et quand il est abonné à la newsletter. J'ai donc créé deux blocs CMS différents dans le BO pour chacun de ces cas. Puis j'ai codé une méthode qui vérifie si le client est premium, auquel cas grâce à une méthode native, on modifie le bloc initialement déclaré par défaut dans le layout et on le remplace par celui du client premium. De la même manière, j'ai créé une méthode pour les abonnés à la newsletter.

FIGURE 2.23 – Appel du bloc CMS dans un template

```

1 <div class="box-premium">
2     <?php if ($this->isPremium()) : ?>
3         <div class="bloc-cms grid-dashboard-bloc-cms">
4             <?php echo $this->getCmsBlockPremiumHtml(); ?>
5         </div>
6     <?php
7     else : ?>
8         <div class="bloc-cms grid-dashboard-bloc-cms">
9             <?php echo $this->getChildHtml('customer_become_premium'); ?>
10        </div>
11    <?php endif?>
12</div>

```

FIGURE 2.24 – Méthode PHP pour appeler le bon bloc CMS si le client est premium

```
1 /**
```

```

2  * Get HTML for CMS block premium
3  *
4  * @return string
5  */
6  public function getCmsBlockPremiumHtml()
7  {
8      $cms = $this->getChild( 'customer_become_premium' );
9
10     $cms->setBlockId( 'customer_already_premium' );
11
12     return $cms->toHtml();
13 }

```

Le principal avantage de cette partie est de m'avoir fait prendre en main la gestion des blocs CMS qui sont souvent présents dans les différents projets car ils permettent au client de contribuer du code eux mêmes et sont donc très souple. J'ai donc appris à les intégrer et à les conditionner, mais aussi à les manipuler depuis les BO qui restait jusqu'à présent un peu flou pour moi car je ne suis pas souvent amené à le configurer.

Gestion des informations clients

Enfin ma dernière mission de développement sur ce projet a été gérer tous les formulaires que l'on peut trouver au niveau du compte client. Cela va de la popin de connexion lorsque l'on est sur une page et que l'on veut se connecter ou s'inscrire, au formulaire de modification des adresses de facturation et de livraison. Je n'ai pas réinventé la roue, j'ai réutilisé les styles déjà développés en début de projet par d'autres collaborateurs. Je me suis assuré que seuls les champs nécessaires s'affichaient, que la vérifications des données côtés utilisateur et serveur s'effectuaient bien et enfin harmoniser le style des pages car elles n'étaient pas toutes présentes dans les maquettes graphiques (cf. Annexe Illustration formulaire de modification des informations personnelles page 55).

J'ai eu l'occasion de refaire des scripts JavaScript et Prototype (un autre framework JavaScript utilisé sur Magento par default). Mon expérience passée sur les autres projets avec ce langage m'ont permis d'être efficace sur ce sujet ce qui m'a laissé du temps pour peaufiner le style des formulaires.

MODIFIER VOS INFORMATIONS

PRÉNOM	<input type="text" value="test"/>					
NOM	<input type="text" value="test"/>					
VOTRE ADRESSE EMAIL	<input type="text" value="hucon@smile.fr"/>					
DATE DE NAISSANCE	<input type="text" value="12"/>	JJ	<input type="text" value="10"/>	MM	<input type="text" value="1990"/>	AAAA
PHOTO D'IDENTITÉ	<input type="button" value="Choisissez un fichier"/> Aucun fichier choisi <small>Seules les photos au format .jpeg, .jpg, .png, .gif sont acceptées. La taille maximum ne doit pas dépasser 2 Mo.</small>					

MOT DE PASSE ACTUEL	<input type="password"/>
NOUVEAU MOT DE PASSE	<input type="password"/>
CONFIRMER NOUVEAU MOT DE PASSE	<input type="password"/>

FIGURE 2.25 – Formulaire de modification des informations client

Bilan du projet

Ce dernier projet fait vraiment office de bilan pour mon stage car il m'a permis de réutiliser toutes les connaissances techniques acquises au cours de ces six mois. En plus j'ai pu apprendre et utiliser des fonctionnalités plus avancées de Magento, donc j'ai encore une fois continué d'enrichir mon bagage techniques dans le domaine du développement Web. Le fait qu'il s'agisse d'un gros projet m'a donné l'occasion d'être plus indépendant sur la partie du site qu'il m'acquittait de développer. Bien entendu je pouvais tout de même compter sur les autres membres du projet pour venir me conseiller en cas de difficulté.

Le fait d'arriver en cours de projet a finalement été intéressant car cela m'a obligé à bien communiquer avec le reste de l'équipe pour être sûr de ne pas réinventer la roue mais plutôt de réutiliser d'éventuels codes déjà réalisés précédemment. De plus le chef de projet réalise des réunions d'équipe régulièrement ce qui permet non seulement de savoir où en est l'avancement du projet mais aussi d'être au courant de la répartition des tâches au différents membres du projet. Ainsi lorsque l'on a une question sur un sujet on sait rapidement à qui s'adresser.

A l'heure où j'écris ces lignes le projet n'est pas encore fini mais il est en bonne voie et devrait être livré avec deux semaines de retard. En effet, pour diverses raisons l'avancement a pris du retard et les responsables du projet en accord avec le client ont décidé de reculer la date de livraison. Encore une fois la principale difficulté est donc, le temps. Pour ma part cette « pression » de planning m'a obligé à être encore plus efficace mais c'est une des réalités du métier il est donc intéressant d'y faire face.

Finalement un des intérêts principal de ce projet a été la réalisation de code de qualité et indépendant, associé à une bonne documentation. Ceci permet aux autres collaborateurs de récupérer le code si besoin et donc de gagner un temps précieux. Afin de vérifier la qualité du code, le chef de projet technique a mis en place un système que l'on intègre à notre IDE et qui vérifie en temps réel que notre code respecte les conventions fixées (indentation, retour à la ligne, commentaires, ...). Ce chef de projet étant un expert sur la technologie, il vérifie régulièrement le code commité et peu nous faire des retours personnalisés quand à la qualité et l'efficacité de notre code. Ce qui se traduit par une progression au niveau de ma connaissance et ma maîtrise de Magento.

Conclusion

Enseignements et bénéfices

Etant issue d'une formation au coeur de métier différent que celui dans lequel j'ai réalisé mon stage, j'ai pris un certain risque avec ce choix. Cependant je savais qu'il s'agissait d'un domaine qui m'intéressait et que je pratiquais déjà en dehors sur mon temps libre. En SRC et tout au cours de l'enseignement à l'INSA j'ai eu de nombreuses heures de programmation dans divers langages et appliqués à divers domaines. Fort heureusement les bases et la logique reste là même tout le temps ce qui m'a permis de m'adapter et d'apprendre rapidement. C'est donc sans regret que je me suis ouvert au domaine du développement Web.

Le fait d'avoir effectué ce stage m'a évidemment permis d'acquérir de nombreuses connaissances techniques et humaines, mais encore plus du fait que celui-ci se soit réalisé dans une entreprise d'un autre secteur. J'ai de nombreuses occasions de me former, que ce soit par moi même, au cours des formations proposées par l'entreprise et surtout grâce au partage entre collaborateur. La notion de travail d'équipe autour d'un projet a véritablement pris tout son sens pendant ce stage, notamment au cours des gros projets sur lesquels j'ai été amené à travailler. Bien que nous ayons eu l'occasion d'effectuer des projets à plusieurs à l'école, le faire en entreprise apporte une autre dimension et qui permet d'acquérir une expérience très enrichissante pour notre carrière professionnelle.

Bien entendu tout n'a pas été rose durant ces six mois, il a fallu apprendre à gérer la pression du planning sur les projets. J'ai rapidement compris que le temps était en quelque sorte notre pire ennemi en entreprise surtout qu'il est obligatoirement lié à des contraintes budgétaires. La notion de budget est-elle aussi importante et nouvelle pour nous qui venons du milieu scolaire. Il faut forcément la prendre en considération lorsque nous développons quitte à parfois se limiter en terme de performance et de qualité de code lorsque le budget ne le permet plus. Comme toutes les entreprises, Smile cherche à générer des profits, on ne peut donc pas aller au delà des ressources budgétaires fournies par le client.

Les projets à venir

Annexes

Annexe Code CSS simulateur d'épargne

```
1 <style type="text/css">
2 body {
3     margin: 0px;
4     padding: 0px;
5     font-family: Arial, Helvetica, sans-serif;
6     font-size: 22px;
7     color: #545454;
8     text-align: center;
9     font-size: 100%;
10    -webkit-print-color-adjust: exact;
11    width: 100%;
12 }
13 h1, h2, h3, h4, h5, h6, p, form, input{
14     margin: 0px;
15     padding: 0px;
16 }
17 a {
18     text-decoration: none ;
19     color: #545454;
20 }
21 a:hover {
22     text-decoration: underline;
23 }
24 #col_centre{
25     width: 100%;
26 }
27 .simulateur{
28     width: 100%;
29 }
30 .simulateur .simulateur-result-left{
31     width: 100%;
32     display: block;
33 }
34 .simulateur .simulateur-result-right{
35     width: 100%;
36     display: block;
37     text-align: center;
38 }
39 .conditions-simulateur{
40     width: 100%;
41     font-size: 18px;
42 }
43 #col_centre .simul .titre h2{
44     font-size: 36px;
45 }
46 .simul h3{
47     font-size: 30px;
48 }
49 .simulateur .input-taux{
50     font-size: 22px;
51     color: #ed6800;
52 }
53 .simulateur .label-right{
54     font-size: 22px;
55 }
56 .simulateur .ligne-simulateur{
57     font-size: 22px;
58     margin-bottom: 15px;
59 }
```



```
60 input, select{
61     text-align: right;
62     font-size: 22px;
63     color: #ed6800;
64 }
65 .input-taux{
66     text-align: left;
67 }
68 </style>
```

FIGURE 2.26 – Code CSS simulateur d'épargne

Annexe Transmission de données d'un formulaire

```

1 <script type="text/javascript">
2   jQuery(document).ready(function() {
3       var currentinput = null;
4       var currentarea = null;
5       var currentvalue = null;
6       jQuery('input').focus(function(){
7           currentinput = jQuery(this).prev('label').text();
8       });
9       jQuery('textarea').focus(function(){
10          currentarea = jQuery(this).prev('label').text();
11      });
12      jQuery('.simple-text + input').blur(function(){
13          currentvalue = jQuery(this).val();
14          jQuery('.hidden label').each(function() {
15              if (currentinput == jQuery(this).text()) {
16                  jQuery(this).next('input').val(currentvalue);
17              }
18          });
19      });
20      jQuery('.text-area + textarea').blur(function(){
21          currentvalue = jQuery(this).val();
22          jQuery('.hidden label').each(function() {
23              if (currentarea == jQuery(this).text()) {
24                  jQuery(this).next('textarea').val(currentvalue);
25              }
26          });
27      });
28      jQuery('.date + input.date-question').blur(function(){
29          jQuery('.hidden .date + input.date-question').val(jQuery(this).val());
30      });
31      jQuery('.email + input').blur(function(){
32          jQuery('.hidden .email + input').val(jQuery(this).val());
33      });
34      jQuery('.numeric + input.input_numeric').blur(function(){
35          currentvalue = jQuery(this).val();
36          jQuery('.hidden label').each(function() {
37              if (currentinput == jQuery(this).text()) {
38                  jQuery(this).next('input').val(currentvalue);
39              }
40          });
41      });
42      jQuery('.civility + ul input').blur(function(){
43          if ( jQuery('.hidden .civility + ul input').length != 0 ) {
44              if (jQuery(this).val() == '1')
45                  jQuery('.hidden #civillite_Mme').attr('checked', true);
46              else {
47                  jQuery('.hidden #civillite_M').attr('checked', true);
48              }
49          } else {
50              jQuery('.hidden .civility + ul input').val(jQuery(this).val());
51          }
52      });
53      jQuery('input').trigger("focus");
54      jQuery('textarea').trigger("focus");
55      jQuery('#existing_product_oui').bind('click', function() {
56          jQuery('input').trigger("focus");
57          jQuery('textarea').trigger("focus");
58          alert("{ /literal }{$node.data_map.popin_message.content}{ literal }");
59          jQuery(this).closest('form').attr("action", "{ /literal }{$node.data_map.redirect_url.content.main_node_id}{ literal }");
60          jQuery(this).closest('form').submit();
61      });
62  });
63 </script>

```

FIGURE 2.27 – Transmission de données d'un formulaire

Annexe Illustrations pages téléconseillers

En poursuivant votre navigation sur notre site, vous acceptez l'utilisation de cookies pour vous proposer des services et contenus adaptés à vos centres d'intérêts et réaliser des statistiques de visites. Pour en savoir plus et paramétrer les traceurs, [cliquez ici](#).

BFM
PROFESSION BANQUIER
VOCATION SOLIDARITÉ

Accueil > Prêt personnel > Prêt -35 ans BFM Energie > Simulation

1 BANQUE DES AGENTS DU SECTEUR PUBLIC

• Trouver votre agence
• Vos comptes en ligne

Compte SG/BFM | **Prêt personnel** | Epargne | Solutions immobilières | Rechercher

Simulation Prêt BFM Energie

Simulateur de prêt

- ▶ Prêt -35 ans BFM Energie
- ▶ Simulation
- Informations complémentaires
- Barème des taux
- Assurance emprunteur
- Convention AERAS

RAPPEL GRATUIT EN MOINS DE 10 MIN

Téléphone Mentions CNIL

INFORMATION COMMERCIALE PAR E-MAIL

☐ Recevoir nos offres

Simulation

Prêt personnel avec un Taux Annuel Effectif Global fixe de 2.29% *

Montant €

Durée mois

Mensualités €

☒ Vous avez moins de 35 ans

Résultats de votre simulation

Montant total du prêt 2 000 €


Durée du prêt 12 mois

BFM Profession banquier Vocation solidarité **BFM et MOI** Nos mutuelles sociétaires

Nous contacter | Presse | Newsletter | FAQ | Nos partenaires | Cookies | Informations légales


moins de 10 minutes
Téléphone Mentions CNIL

FIGURE 2.28 – Pages téléconseillers popin simulation



PROFESSION BANQUIER
VOCATION SOLIDARITÉ

Accueil > Page Simulateurs



- Trouver votre agence
- Vos comptes en ligne

BANQUE FRANÇAISE MUTUALISTE

Compte SG/BFM | Prêt personnel | Epargne | Solutions immobilières |

Page Simulateurs

Page Simulateurs

Choix du prêt *

Prêt -35 ans

Montant *

2 000 €

Durée *

12 mois

SIMULER

Adresse générée

<http://www.bfm.fr/Pret-personnel/Vous-avez-Moins-de-35-ans-Empruntez-a-taux-attractif/Simulation-Pret-BFM-Energie?duree=12&montant=2000>

Civilité *

☐ Mme ☒ M

Nom *

Test Nom

Prénom *

Test Prénom

Adresse e-mail *

test@test.com

Choix du prêt *

Prêt -35 ans

URL du simulateur *

<http://www.bfm.fr/Pret-personnel/Vous-avez-Moins-de-35-ans-Empruntez-a-tau>

Durée *

12

Montant *

2 000

Champs obligatoires(*)

Envoyer

Recevoir nos offres

BFM

Profession banquier

Vocation solidarité

BFM et MOI

Nos mutuelles sociétaires

Nous contacter

Presse

Newsletter

FAQ

Nos partenaires

Cookies

Informations légales

Des informations commerciales?
Nous vous rappelons en moins de 10 minutes

Mentions CNIL

FIGURE 2.29 – Pages téléconseillers formulaire préremplie

Annexe Illustration header Darjeeling



FIGURE 2.30 – Header Darjeeling

Annexe Installeur Magento pour ajouter un attribut au BO

```

1 <?php
2 /**
3  * Update type 360 attribute
4  *
5  * @category Darjeeling
6  * @package Darjeeling_Core
7  * @copyright 2013 Darjeeling
8  */
9  /* @var $this Smile_DarjeelingCore_Model_Resource_Eav_Mysql4_Setup */
10 $installer = $this;
11 $installer->startSetup();
12
13 $entityTypeId = $installer->getEntityTypeId('catalog_category');
14 $attributeSetId = $installer->getDefaultAttributeSetId($entityTypeId);
15 $attributeGroupId = $installer->getDefaultAttributeGroupId($entityTypeId, $attributeSetId);
16
17 $installer->addAttribute('catalog_category', 'promo_img_link', array(
18     'type' => 'varchar',
19     'label' => 'Link for promotion image',
20     'input' => 'text',
21     'global' => Mage_Catalog_Model_Resource_Eav_Attribute::SCOPE_STORE,
22     'visible' => true,
23     'required' => false,
24     'user_defined' => false
25 ));
26
27 $attributeId = $installer->getAttributeId($entityTypeId, 'promo_img_link');
28
29 $installer->run("
30 INSERT INTO `{$installer->getTable('catalog_category_entity_varchar')}`
31 (`entity_type_id`, `attribute_id`, `entity_id`, `value`)
32 SELECT `{$entityTypeId}`, `{$attributeId}`, `entity_id`, NULL
33 FROM `{$installer->getTable('catalog_category_entity')}`";
34 );
35
36
37 //this will set data of your custom attribute for root category
38 Mage::getModel('catalog/category')
39     ->load(1)
40     ->setImportedCatId(0)
41     ->setInitialSetupFlag(true)
42     ->save();
43
44 //this will set data of your custom attribute for default category
45 Mage::getModel('catalog/category')
46     ->load(2)
47     ->setImportedCatId(0)
48     ->setInitialSetupFlag(true)
49     ->save();
50
51 $installer->endSetup();

```

FIGURE 2.31 – Installeur Magento pour ajouter un attribut au BO

Annexe Script jQuery d'ajustement d'une info bulle

```

1 <script>
2 jQuery(document).ready(function () {
3     jQuery( '.express-wishlist input.add-to-wishlist ' ).each(function() {
4         jQuery( this ).attr("title", '<?php echo $this->__("Add to my cravings"); ?>');
5     });
6     <?php if ( $this->helper('customer')->isLoggedIn()):?>
7         <?php $_wishlist = Mage::helper('darjeelingdesign/customer')->
            getCustomerWishlistItemsToJavascript(); ?>
8         <?php if (!empty($_wishlist)): ?>
9             /* Display Wishlist Items */
10            <?php $wishlist = explode(" ", $_wishlist); ?>
11            <?php foreach($wishlist as $wish): ?>
12                jQuery("<?php echo $wish ?>").addClass('wished');
13                jQuery("<?php echo $wish ?>").attr("title", '<?php echo $this->__("Remove from my
                    cravings"); ?>');
14            <?php endforeach; ?>
15        <?php endif; ?>
16    <?php endif;?>
17    jQuery( '.tooltip ' ).tooltipster();
18    jQuery( '.express-wishlist input.add-to-wishlist ' ).each(function() {
19        jQuery( this ).attr("title", '<?php echo $this->__(""); ?>');
20    });
21 });
22
23 jQuery( '.express-wishlist input.add-to-wishlist ' ).click( function() {
24     if( jQuery( this ).hasClass( 'wished' ) ) {
25         jQuery( this ).tooltipster('content', '<?php echo $this->__("Add to my cravings"); ?>');
26     }
27     else {
28         jQuery( this ).tooltipster('content', '<?php echo $this->__("Remove from my cravings"); ?>');
29     }
30 });
31 </script>

```

FIGURE 2.32 – Script jQuery d'ajustement d'une info bulle

Annexe Illustration voir plus de produits Darjeeling

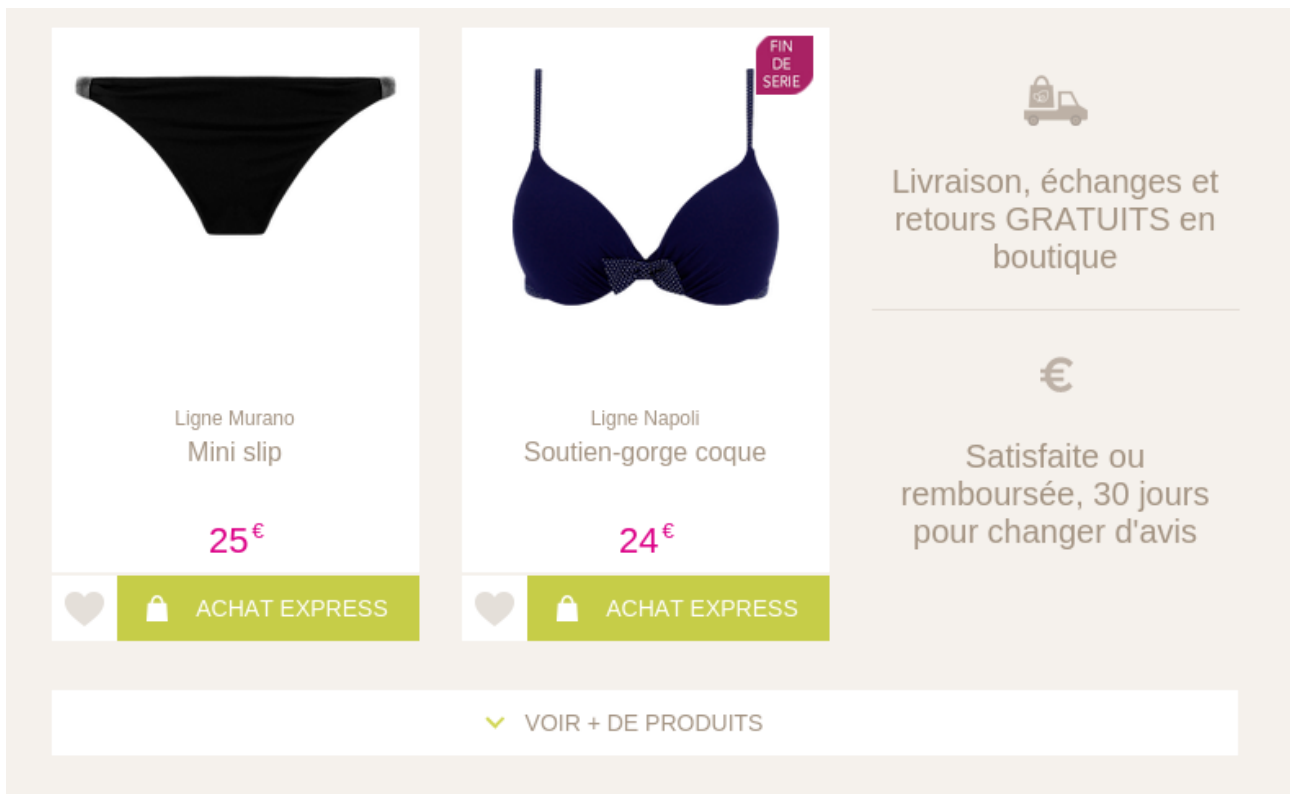


FIGURE 2.33 – Bloc pour voir plus de produits

Annexe Illustration dashboard Snowleader

The screenshot displays the Snowleader client dashboard. The top navigation bar includes links for 'RETOURS GRATUITS', 'LIVRAISON DÈS 1€', and 'UNE QUESTION ? NOUS SOMMES LÀ'. The left sidebar features categories like SNOW, OUTDOOR, URBAN, and MARQUES, along with 'Tendances', 'Conseils', and 'Bons plans'. The main content area is divided into several sections:

- MON COMPTE**: A section for account management. It includes a greeting 'BONJOUR, TEST TEST!' and a note 'Vous n'êtes pas encore client Premium ?'. It displays the user's email 'hucon@smile.fr' and a default billing address in France. Buttons for 'MES INFORMATIONS' and 'MON CARNET D'ADRESSES' are present.
- VOS POINTS ET CRÉDITS**: A section showing loyalty points and credits. It indicates 215 points (worth 215.00 €) and 160 credits (worth 160.00 €). A 'DETAILS' button is available.
- MES COMMANDES**: A section for viewing order history. It lists two orders from July 24, 2015, both in 'COMMANDE EN PRÉPARATION' status. The first order (Ref: 100233859) is for 2 articles worth 289.59 €. The second order (Ref: 100233858) is for 5 articles worth 619.01 €.

FIGURE 2.34 – Dashboard client Snowleader

Annexe Condition sur le lien d'impression de facture

```

1 <div class="button-print-wrapper">
2   <?php if( $_order->hasInvoices() ) : ?> // Verification of existing invoice
3     <a href="<?php echo $this->getPrintAllInvoicesUrl($_order) ?>" onclick="this.target='_blank';" title="<?php echo $this->__('Print the bill'); ?>"></a> // Link to invoice print page
4   <?php else : ?> // In case of no existing invoice
5     <a href="<?php echo $this->getPrintUrl($_order) ?>" onclick="this.target='_blank';" title="<?php echo $this->__('Print the order'); ?>"></a> // Link to order print page
6   <?php endif; ?>
7 </div>

```

FIGURE 2.35 – Condition sur le lien d'impression de facture

Annexe Illustration back-office client Magento

Magento Enterprise Enregistrement global des recherches Connecté en tant que [smb](#) | jeudi 13 août 2015 | [Se déconnecter](#)

Tableau de bord Ventes Catalogue Mobile **Clients** Promotions Bulletin d'information Interacting Slider Concours CMS Blog Rapports Système Magic Toolbox FIA-NET [Obtenir de l'aide pour cette page](#)

Message : Votre serveur web n'est pas configuré correctement. Par conséquent, les fichiers de configuration contenant des informations sensibles sont accessibles de l'extérieur. Veuillez contacter votre hébergeur.

Dernier message : Enterprise Edition 1.14.2.1 is Now Available: Includes Enhanced Security and Performance [Lire détails](#)

Un des index ou plus n'est pas à jour : Index de recherche catalogue. Cliquez ici pour aller à la [Gestion des index](#) et reconstruire les index requis. Vous avez 6 critique et 2 notice message(s) non lues. [Aller à la boîte de réception](#)

Information sur le client Hugo Contreras [Retour](#) [Réinitialiser](#) [Créer une commande](#) [Supprimer le client](#) [Gérer le panier](#) [Enregistrer le client](#) [Sauvegarder et continuer l'édition](#)

Voir le client

- Account Information
- Adresses
- Commandes
- Accords de facturation
- Profils récurrents (beta)
- RMA
- Payment Information
- Panier
- Liste de cadeaux
- Bulletin d'information
- Avis sur le produit
- Étiquettes des produits
- Crédit Magasin
- Registre de cadeaux
- Points de fidélité

Information personnelle

Dernière connexion en :	Jamais (hors ligne)	Adresse de facturation par défaut
E-mail confirmé :	Confirmer	Résidence Eurofac Tour 6
Compte créé sur :		33170 - Gradignan
Compte créé sur (Europe/Paris):	24 juil. 2015 13:39:11	France
Compte créé dans :	Store FR	
Groupe de clients :	Premium	

Solde Points de Fidélité

Site internet	Solde	Montant devises	Solde minimum de points fidélité requis pour pouvoir échanger	Plafonner points de fidélité
Website B2C FR	215	430.00 €	0	Illimité

Solde du crédit magasin

Solde	Site web
175.00 €	Website B2C FR

Statistiques de vente

Site	Magasin	Vue du magasin	Ventes à vie	Vente moyenne
Website B2C FR	Store B2C FR	Store FR	908,60 €	454,30 €
Toutes les vues de la boutique			908,60 €	454,30 €

Commandes récentes

- Panier - 4 article(s)
- Liste de vœux - 0 objets

FIGURE 2.36 – Back-office client Magento

Annexe Déclaration du controller custom loyalty

```

1 <?php
2 /**
3  * Customer Loyalty Controller
4  *
5  * @category Snowleader
6  * @package  Snowleader_Customer
7  */
8 class Snowleader_Customer_LoyaltyController extends Mage_Core_Controller_Front_Action
9 {
10     /**
11      * Only logged in users can use this functionality ,
12      * this function checks if user is logged in before all other actions
13      *
14      */
15     public function preDispatch()
16     {
17         parent::preDispatch();
18
19         if (!Mage::getSingleton('customer/session')->authenticate($this)) {
20             $this->setFlag('', 'no-dispatch', true);
21         }
22         if (!Mage::helper('enterprise_reward')->isEnabledOnFront()) {
23             $this->norouteAction();
24             $this->setFlag('', self::FLAG_NO_DISPATCH, true);
25         }
26     }
27
28     /**
29      * Customer Loyalty Page
30      *
31      */
32     public function indexAction()
33     {
34         /* Gift Card */
35         $data = $this->getRequest()->getPost();
36         if (isset($data('giftcard_code'))) {
37             $code = $data('giftcard_code');
38             try {
39                 if (!Mage::helper('enterprise_customerbalance')->isEnabled()) {
40                     Mage::throwException($this->__('Redemption functionality is disabled. '));
41                 }
42                 Mage::getModel('enterprise_giftcardaccount/giftcardaccount')->loadByCode($code)
43                     ->setIsRedeemed(true)->redeem();
44                 Mage::getSingleton('customer/session')->addSuccess(
45                     $this->__('Gift Card "%s" was redeemed.', Mage::helper('core')->htmlEscape($code))
46                 );
47             } catch (Mage_Core_Exception $e) {
48                 Mage::getSingleton('customer/session')->addError($e->getMessage());
49             } catch (Exception $e) {
50                 Mage::getSingleton('customer/session')->addException($e, $this->__('Cannot redeem Gift Card. '));
51             }
52             $this->_redirect('*/*/*');
53             return;
54         }
55
56         /*Customer Balance */
57         if (!Mage::helper('enterprise_customerbalance')->isEnabled()) {
58             $this->_redirect('customer/account/');
59             return;
60         }
61         $this->loadLayout();
62         $this->_initLayoutMessages('customer/session');
63         $this->loadLayoutUpdates();
64         $headBlock = $this->getLayout()->getBlock('head');
65         if ($headBlock) {
66             $headBlock->setTitle($this->__('Customer Loyalty'));
67         }
68         $this->renderLayout();
69     }
70
71     /**

```

```

72  * Info Action
73  */
74  public function infoAction()
75  {
76      Mage::register('current_reward', $this->_getReward());
77      $this->loadLayout();
78      $this->_initLayoutMessages('customer/session');
79      $headBlock = $this->getLayout()->getBlock('head');
80      if ($headBlock) {
81          $headBlock->setTitle(Mage::helper('enterprise_reward')->__( 'Reward Points' ));
82      }
83      $this->renderLayout();
84  }
85
86  /**
87   * Save settings
88   */
89  public function saveSettingsAction()
90  {
91      if (!$this->_validateFormKey()) {
92          return $this->_redirect('*/*/info');
93      }
94
95      $customer = $this->_getCustomer();
96      if ($customer->getId()) {
97          $customer->setRewardUpdateNotification($this->getRequest()->getParam('subscribe_updates'
98              ))
99              ->setRewardWarningNotification($this->getRequest()->getParam('subscribe_warnings'));
100      $customer->getResource()->saveAttribute($customer, 'reward_update_notification');
101      $customer->getResource()->saveAttribute($customer, 'reward_warning_notification');
102
103      $this->_getSession()->addSuccess(
104          $this->__( 'The settings have been saved.' )
105      );
106      $this->_redirect('*/*/info');
107  }
108
109  /**
110   * Unsubscribe customer from update/warning balance notifications
111   */
112  public function unsubscribeAction()
113  {
114      $notification = $this->getRequest()->getParam('notification');
115      if (!in_array($notification, array('update', 'warning'))) {
116          $this->_forward('noroute');
117      }
118
119      try {
120          /* @var $customer Mage_Customer_Model_Session */
121          $customer = $this->_getCustomer();
122          if ($customer->getId()) {
123              if ($notification == 'update') {
124                  $customer->setRewardUpdateNotification(false);
125                  $customer->getResource()->saveAttribute($customer, 'reward_update_notification'
126                      );
127              } elseif ($notification == 'warning') {
128                  $customer->setRewardWarningNotification(false);
129                  $customer->getResource()->saveAttribute($customer, 'reward_warning_notification'
130                      );
131              }
132              $this->_getSession()->addSuccess(
133                  $this->__( 'You have been unsubscribed.' )
134              );
135          } catch (Exception $e) {
136              $this->_getSession()->addError($this->__( 'Failed to unsubscribe.' ));
137          }
138
139          $this->_redirect('*/*/info');
140      }
141
142  /**
143   * Retrieve customer session model object
144   *
145   * @return Mage_Customer_Model_Session

```

```

145     */
146 protected function _getSession()
147 {
148     return Mage::getSingleton('customer/session');
149 }
150
151 /**
152  * Retrieve customer session model object
153  *
154  * @return Mage_Customer_Model_Session
155  */
156 protected function _getCustomer()
157 {
158     return $this->_getSession()->getCustomer();
159 }
160
161 /**
162  * Load reward by customer
163  *
164  * @return Enterprise_Reward_Model_Reward
165  */
166 protected function _getReward()
167 {
168     $reward = Mage::getModel('enterprise_reward/reward')
169         ->setCustomer($this->_getCustomer())
170         ->setWebsiteId(Mage::app()->getStore()->getWebsiteId())
171         ->loadByCustomer();
172     return $reward;
173 }
174 }

```

FIGURE 2.37 – Déclaration du controller custom loyalty

Annexe Illustration formulaire de modification des informations personnelles

MODIFIER VOS INFORMATIONS

PRÉNOM	<input type="text" value="test"/>				
NOM	<input type="text" value="test"/>				
VOTRE ADRESSE EMAIL	<input type="text" value="hucon@smile.fr"/>				
DATE DE NAISSANCE	<input type="text" value="12"/>	JJ	<input type="text" value="10"/>	MM	<input type="text" value="1990"/> AAAA
PHOTO D'IDENTITÉ	<div>Choisissez un fichier Aucun fichier choisi</div> <div>Seules les photos au format .jpeg, .jpg, .png, .gif sont acceptées. La taille maximum ne doit pas dépasser 2 Mo.</div>				

MOT DE PASSE ACTUEL	<input type="password"/>
NOUVEAU MOT DE PASSE	<input type="password"/>
CONFIRMER NOUVEAU MOT DE PASSE	<input type="password"/>

ENREGISTRER

◀ RETOUR

FIGURE 2.38 – Formulaire de modification des informations personnelles

Annexe Architecture logicielle MVC (Modèle Vue Contrôleur)

Le patron d'architecture logicielle modèle-vue-contrôleur, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Modèle

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue. Sa communication avec la vue s'effectue au travers du patron Observateur.

Le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir tous les cours d'une salle, tous les cours d'une personne ou tous les cours d'un groupe de TD.

Vue

Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Plusieurs vues peuvent afficher des informations partielles ou non d'un même modèle. Par exemple si une application de conversion de base a un entier comme unique donnée, ce même entier peut être affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir à l'utilisateur la possibilité de changer de vue.

Contrôleur

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est lui « observable ». Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser

tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Source Wikipedia

Table des figures

2.1 Nouveaux sites mobiles (2 pour la belgique et 1 pour la russie)	10
2.2 Récupération des données depuis le back-office et affichage de la liste déroulante	13
2.3 Chargement en AJAX du simulateur de prêt	13
2.4 Configuration serveur pour la réécriture des requêtes HTTPS en HTTP	14
2.5 Simulateur global de prêt	15
2.6 Simulateur d'épargne	16
2.7 Landing page	17
2.8 Récupération des questions existantes sur l'autre formulaire	17
2.9 Pages téléconseillers formulaire	19
2.10 Structure de base d'une page Magento	22
2.11 Theme responsive Magento 1.14 demo	23
2.12 Nouveau menu de navigation Darjeeling	25
2.13 Récupération liens à afficher dans le footer depuis le back-office	25
2.14 Footer Darjeeling	26
2.15 Page « Mes envies » Darjeeling	26
2.16 Liste produits	27
2.17 Script jQuery d'affichage en accordéon des filtres	27
2.18 Adaptation du nouveau design sur mobile	29
2.19 Page d'historique des commandes	32
2.20 Détail d'une commande	33
2.21 Nouvelle page Fidélité	34
2.22 Déclaration d'un layout avec l'appel à un bloc CMS	35
2.23 Appel du bloc CMS dans un template	35
2.24 Méthode PHP pour appeler le bon bloc CMS si le client est premium	35
2.25 Formulaire de modification des informations client	37
2.26 Code CSS simulateur d'épargne	41
2.27 Transmission de données d'un formulaire	42
2.28 Pages téléconseillers popin simulation	43
2.29 Pages téléconseillers formulaire préremplie	44
2.30 Header Darjeeling	45
2.31 Installeur Magento pour ajouter un attribut au BO	46
2.32 Script jQuery d'ajustement d'une info bulle	47
2.33 Bloc pour voir plus de produits	48
2.34 Dashboard client Snowleader	49
2.35 Condition sur le lien d'impression de facture	50
2.36 Back-office client Magento	51
2.37 Déclaration du controller custom loyalty	54
2.38 Formulaire de modification des informations personnelles	55