



Agence Web et SEO spécialisée dans le E-commerce

12 rue Bégand, Troyes

Développement Web Intégration

Tanguy Crepy

Rapport de Stage

Département MMI Troyes

Tuteur en entreprise : Benjamin Pescheux

Tuteur académique : David Annebicque

Année 2016-2017

Stage du 02/05/17 au 22/06/17



Remerciements

Je tiens à remercier Benjamin Pescheux, mon tuteur de stage, qui a répondu à mes questions et m'a accompagné durant tout mon stage dans la société ScaleDev.

Je remercie également Thomas Hérisson, Gwennael Dolat, Joris Plomion, et Maxime Michaut pour m'avoir accueilli et conseillé pendant ces 2 mois.

Je souhaite remercier également mon tuteur universitaire, David Annebicque, et tous mes professeurs pour les connaissances et le savoir-faire transmis au cours de ces 2 années de formation.

Sommaire

Introduction	1
Missions principales	3
Mise à jour de la partie « Mon Compte » du site scaledev.fr	4
Analyse de PrestaShop et du site	5
Extension de mise à jour et de support	7
Gestion de l'extension de MAJ dans le Back-office	11
Refonte graphique de la partie « Mon Compte »	12
Mise à jour du site topconsommables.com	14
Analyse de l'import	15
Fusion des stocks sachet et cartouche et import des papiers	16
Intégration des pages produit	19
Panier intermédiaire et cross-selling	20
Missions secondaires	23
Conclusion	26
Glossaire	28
Résumé	30
Abstract	30

Introduction



Commençons par présenter l'entreprise ScaleDev, qui est une agence web e-commerce spécialisée dans le développement sur PrestaShop.

L'entreprise a été créée en 2009 à Nice par Thomas Herisson sous le nom de CreaWeb06. Par la suite l'agence est devenue ScaleDev et s'est spécialisée dans le e-commerce.

Elle propose de nombreux services, comme la vente de module PrestaShop, la création et le suivi de boutiques des clients, ou encore la location de serveur web.

Implantée à Troyes, elle traite avec des clients à l'échelle internationale.

Thomas Herisson
Directeur de l'agence

Benjamin Pescheux
Chef de projet
Responsable Développement

Joris Plomion
Développeur web

Gwennael Dolat
Responsable Pôle
SEO et Marketing

Maxime Michaut
Développeur web

<!--Composition de l'équipe-->

Les membres de l'équipe travaillent sur de nombreux projets et effectuent un suivi de leurs clients en répondant à leurs demandes le plus rapidement possible.

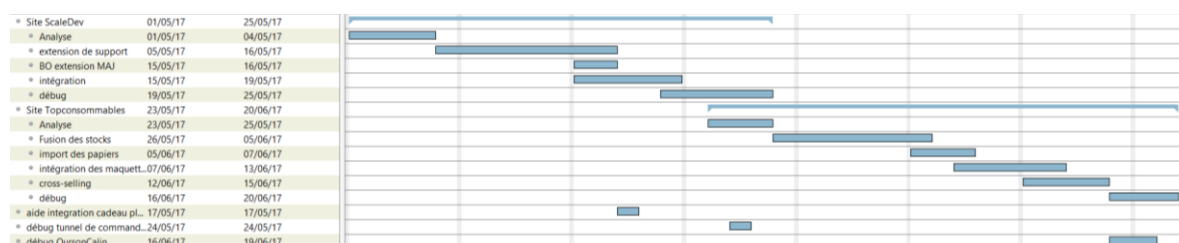
Durant ce stage, mon tuteur, Benjamin, m'a donné différentes tâches à accomplir. J'ai donc eu l'occasion de travailler sur deux missions principales, mais aussi d'intervenir sur des problèmes secondaires dans le cadre de ce stage ayant débuté le 02/05/17 et finissant le 22/06/17.

Ma première mission consistait à mettre à jour le site scaledev.fr, et plus particulièrement la partie « Mon Compte ».

Ma deuxième mission était de modifier la tâche d'import des produits et d'intégrer les différentes fiches produits sur le site topconsommables.com.

Chacune des missions travaillées concernaient un site basé sur le CMS PrestaShop, et étaient liées au développement backend, frontend et à l'intégration de maquettes graphiques.

Ce rapport vous décrira la façon dont ces huit semaines de stage se sont déroulées. Je commencerai par vous expliquer les deux missions principales que j'ai eu à traiter, puis les différentes missions secondaires sur lesquelles j'ai pu travailler. Pour conclure, Je vous parlerai des connaissances et de l'enrichissement personnel que ce stage m'a apporté.

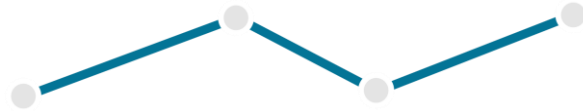


<!--Diagramme de Gantt-->

Missions principales



Mise à jour de la partie « Mon Compte » du site scaledev.fr



Lors de mon arrivée dans l'entreprise, je ne connaissais que très peu le CMS PrestaShop. Grâce aux cours de MMI, je savais comment construire une requête SQL et afficher son résultat dans PrestaShop, mais cela ne restait qu'une approche de ce CMS. Pour m'initier à sa structure et à son fonctionnement, Benjamin a commencé par me faire travailler sur la partie « Mon Compte » du site ScaleDev. Cette section nécessitait le développement d'une nouvelle fonctionnalité, l'intégration d'un nouveau design, et la correction de divers bugs.

En complément de ce travail, j'ai fait de nombreuses recherches sur les forums et dans les documentations des technologies utilisées afin de comprendre le fonctionnement du CMS.

Analyse de PrestaShop et du site

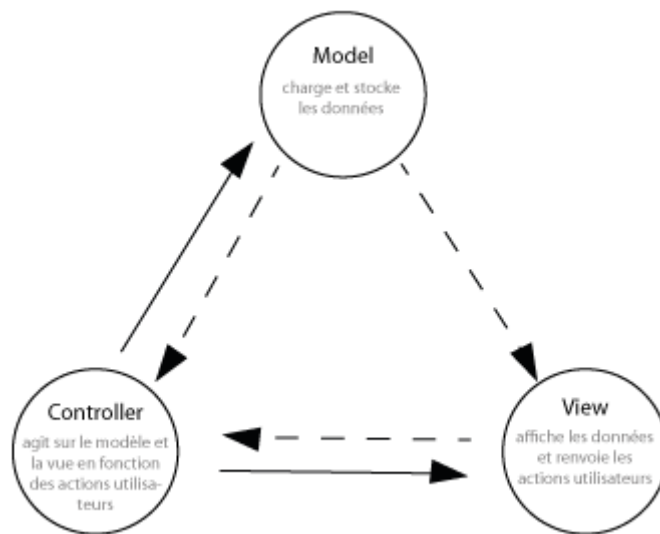
Le site de l'agence propose la vente de module pour le CMS PrestaShop, et pour chaque module vendu, propose également une période de mise à jour et de support choisie lors de l'achat du module. Jusqu'à présent, les clients ne pouvaient étendre cette période de mise à jour et de support qu'en contactant l'agence par mail ou téléphone. Une de mes problématiques était donc de mettre en place un système permettant l'extension de cette période.

Mon travail a commencé par l'analyse du code et de la structure du site. Prestashop est un CMS qui se base sur une structure Model View Controller. Cette structure est utilisée par de nombreuses applications, car elle permet de séparer les différents éléments du code.

Les models (Classes dans PrestaShop) permettent la liaison avec la base de données grâce à de nombreuses fonctions. Dans ces fichiers, on utilisera principalement des requêtes SQL.

Les views sont regroupées dans un dossier « thèmes ». Les thèmes de PrestaShop sont séparés en fichier JavaScript, CSS, et HTML. Les fichiers HTML sont en fait des fichiers terminant par « .tpl » et fonctionnent sur un moteur de template appelé Smarty. Ce moteur de template permet l'utilisation de diverses fonctions propre à Smarty tout en permettant une bonne lisibilité du code. Mais l'avantage premier d'un moteur de template est l'héritage de template entre eux (Le template est en fait l'architecture du design d'une page. En quelques sortes la trame de fond.). Chaque bloc affiché aura son fichier tpl (extension de fichier, tpl pour template) correspondant et s'intégrera dans un fichier parent correspondant à la structure principale de la page.

Enfin les controllers permettent de faire la liaison entre les views et les models. Ils permettent de récupérer le résultat des fonctions du model, de les traiter, et de les envoyer dans la view que l'on affiche.






<!--Schéma théorique MVC-->

De plus PrestaShop possède des fichiers de base, qu'il faut éviter de modifier si l'on veut pouvoir mettre à jour le site. Pour pallier à ce problème, un système a été mis en place. Dans un dossier Override, on place des fichiers avec le même nom que le fichier initial qui vont hériter de ce dernier. On parle alors d'override d'un fichier.

Extension de mise à jour et de support

Une fois cette analyse terminée, j'ai pu commencer à intégrer l'extension de mise à jour pour les modules vendus par ScaleDev. J'ai commencé par intégrer 3 boutons correspondant à des périodes de 3 mois, 1 an et 2 ans. J'ai créé un produit extension de licence avec 3 déclinaisons pour chaque module vendu par l'agence dans le back-office.

<input type="checkbox"/>	41		Extension support : Monetico	Services	0,00 €	46,80 €	✓	Modifier ▼
<input type="checkbox"/>	42		Extension support : Manomano	Services	0,00 €	94,80 €	✓	Modifier ▼
<input type="checkbox"/>	43		Extension support : Be2Bill	Services	0,00 €	46,80 €	✓	Modifier ▼

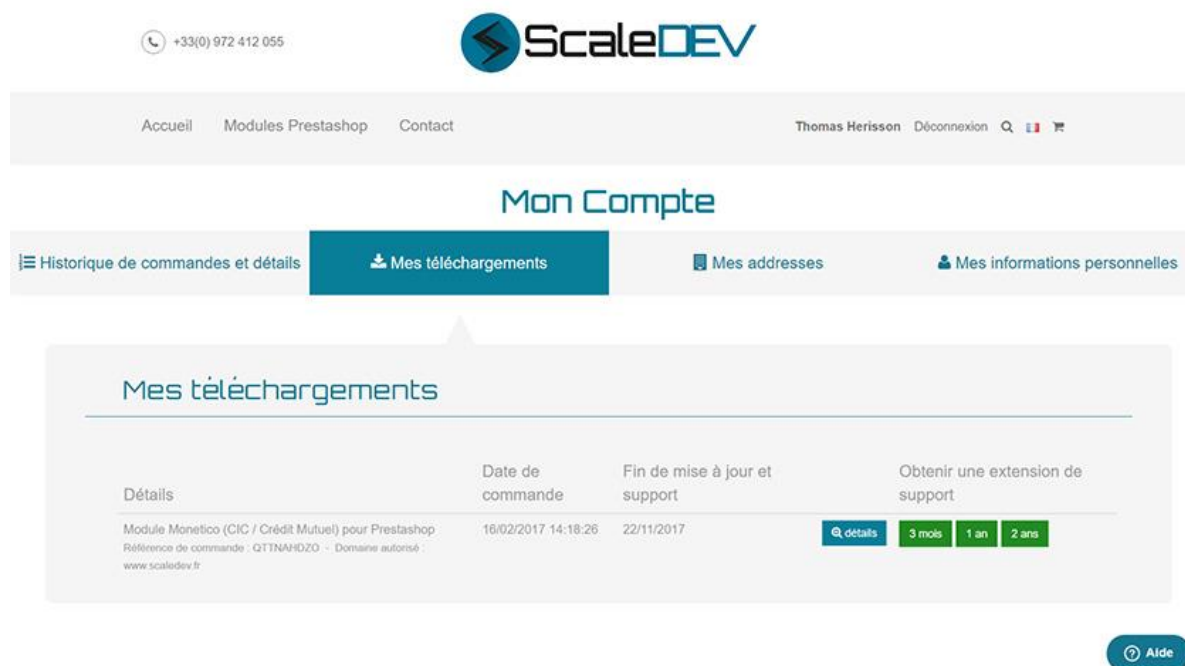
<!--Produits « Extension de licence » dans le back-office-->

Ma première difficulté a été de comprendre comment un ajout au panier fonctionnait. J'ai donc analysé les différentes pages produit et j'ai pu isoler le lien d'ajout au panier.

Ce dernier est construit grâce à la fonction `getPageLink` de l'objet `$link`. Les paramètres qui lui sont passés sont le controller que l'on veut appeler, ici le controller `Cart`, ainsi que les variables envoyées au controller pour être traitées, ici l'ID (identifiant unique d'un enregistrement dans une base de données) du produit, l'ID de l'attribut du produit (3 mois, 1 an, 2 ans) et sa quantité.

En suivant les conseils de Benjamin, j'ai créé une table intermédiaire afin de lier les extensions de licence à leur module complémentaire, dans laquelle est enregistré l'ID du produit d'extension de licence et l'ID du module qui lui correspond. Cela permet de récupérer le bon produit en fonction du module acheté par le client.

Ceci étant fait, j'ai pu intégrer correctement les boutons dans la page « Mes téléchargements ».



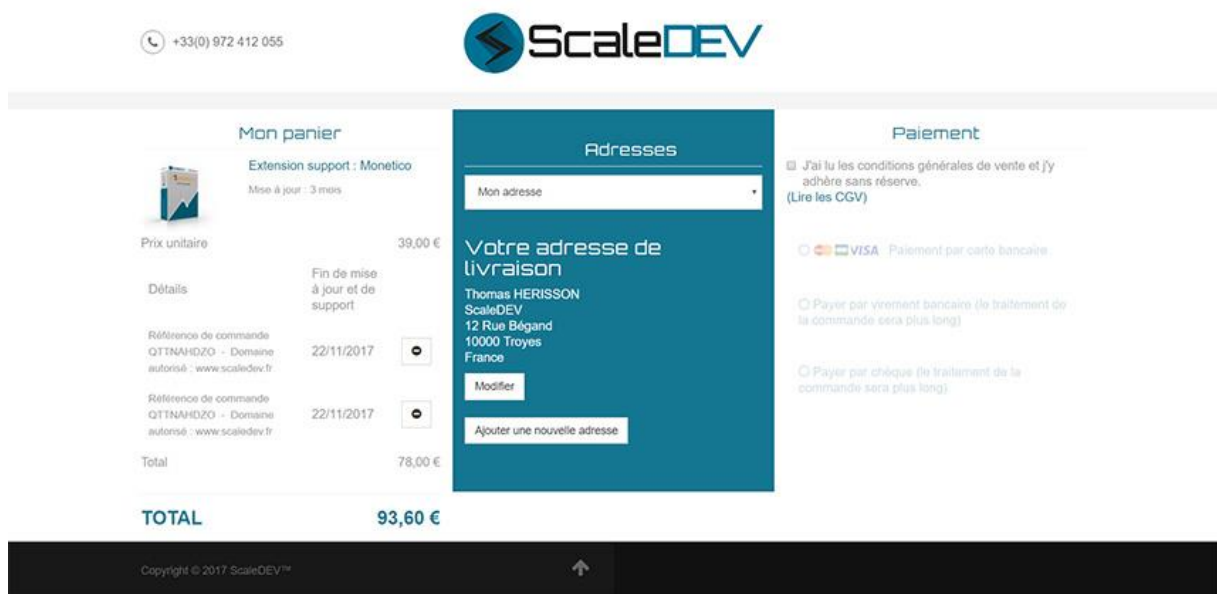
<!--Page « Mes Téléchargements »-->

Une nouvelle problématique est alors apparue. Chaque module vendu est lié à un contrat, mais les produits qui ont été créés plus tôt ne permettait pas d'être lié à ces contrats automatiquement mais seulement à un module. Pour résoudre ce problème, on envoie l'ID du module correspondant à l'extension de licence au controller du panier qui va se charger d'enregistrer cet ID dans un tableau sérialisé, c'est à dire un tableau qui a été transformé en chaîne de caractère grâce à une fonction PHP, qui sera envoyé dans la table « ps_cart_product » de la base de données.

Dans cette table il y a le produit qui a été ajouté au panier, et sa quantité. Grâce au tableau, pour chaque quantité ajoutée ou supprimée, on ajoute ou supprime l'ID du contrat associé.

Les extensions de licence sont maintenant correctement ajoutées au panier, et sont bien liées au contrat qui leur correspond. Mais une fois dans la page panier, elles s'affichent comme n'importe quel autre produit, avec une image du produit et des boutons pour modifier leur quantité.

Il fallait afficher en plus du nom du produit, les contrats auxquels le produit était lié sous la forme d'une liste. J'ai réfléchi à la meilleure manière d'afficher cette liste, et suite à cette réflexion, j'ai désigné l'affichage des extensions de licence dans le panier. Les contrats s'affichent comme dans la page « Mes téléchargements », afin d'éviter de perdre le client.



<!--Panier avec différents produit sélectionné-->

De plus, il fallait empêcher le client de pouvoir augmenter la quantité d'extensions dans le panier, sans quoi il y aurait eu des extensions sans contrat lié.

J'ai donc ajouté un bouton de suppression du produit qui permet de décrémenter de 1 la quantité de l'extension, et qui prend en paramètre l'ID du contrat lié qui est retiré du tableau enregistré dans la base de données. Le reste des boutons de gestion de la quantité du produit ont été supprimés pour ce type de produits.

Le bouton de suppression est placé au bout de chaque ligne du tableau des contrats concernés par l'extension de licence pour bien montrer quel contrat sera effacé de la liste.

Gestion de l'extension de MAJ dans le Back-office

Enfin, il ne reste plus que le traitement de l'extension ajoutée au panier lors du changement de statut de la commande passant de « en attente de paiement » à « paiement accepté ». Cela se passe dans le Hook de changement de statut de la commande.

The screenshot shows a web interface for managing orders. At the top, there are two tabs: 'ÉTAT' with a count of 2 and 'DOCUMENTS' with a count of 0. Below the tabs is a table of orders. The first row is highlighted in light blue and shows a status of 'Païement accepté' (Payment accepted) with a green checkmark icon, the name 'Scale DEV', and the timestamp '19/06/2017 09:41:12'. To the right of this row is a button labeled 'Renvoyer l'e-mail'. The second row shows a status of 'En attente du paiement par chèque' (Waiting for payment by check) with a checkmark icon, the name 'Scale DEV', and the timestamp '19/06/2017 09:40:38'. To the right of this row is also a button labeled 'Renvoyer l'e-mail'. Below the table is a form with a dropdown menu currently set to 'Païement accepté' and a blue button labeled 'METTRE À JOUR L'ÉTAT'.

Statut	Nom	Date	Action
Païement accepté	Scale DEV	19/06/2017 09:41:12	Renvoyer l'e-mail
En attente du paiement par chèque	Scale DEV	19/06/2017 09:40:38	Renvoyer l'e-mail

Païement accepté METTRE À JOUR L'ÉTAT

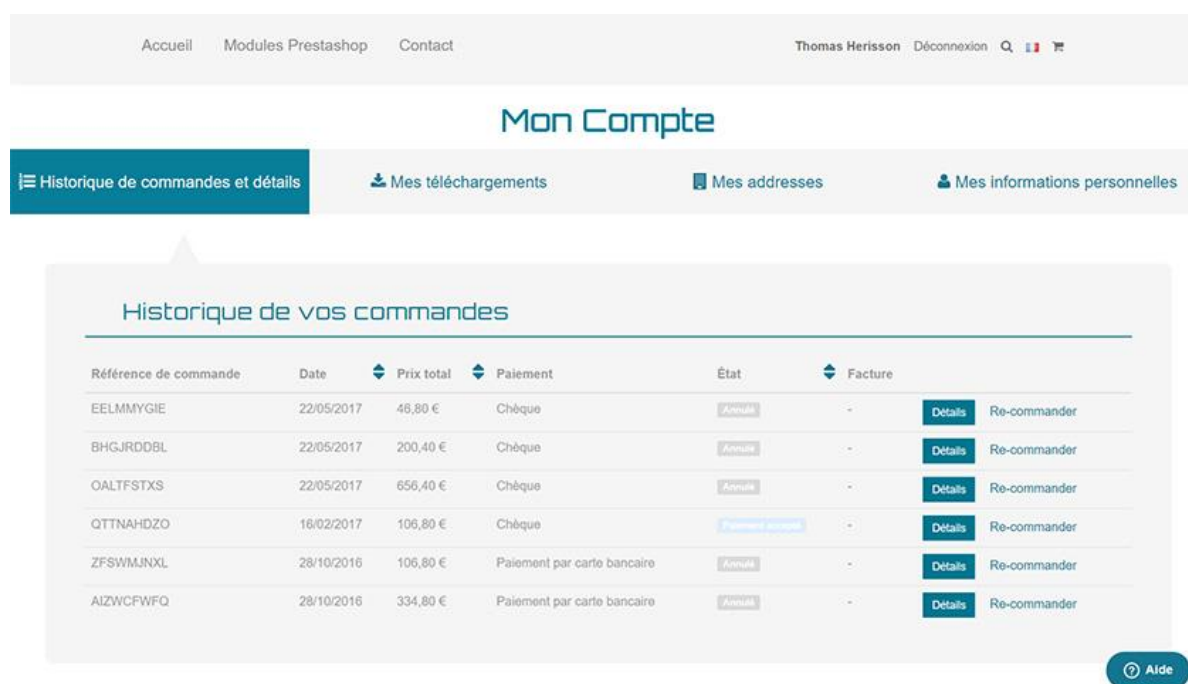
<!--Bloc changement de statut BO-->

Si le produit payé est une extension de licence, on récupère les ID des contrats contenus dans le tableau sérialisé de la base de données. Grâce à ces ID, on peut obtenir les contrats et leurs détails, et plus précisément la date de fin de mise à jour et de support. Si celle-ci est postérieure à la date du jour, on lui ajoute le temps sélectionné par le client (3 mois, 1 an ou 2 ans). Sinon on utilise la date du jour pour cela.

Il ne restait plus qu'à afficher la date de fin de support dans la liste des commandes dans le back-office. Pour cela, j'ai modifié le HookDisplayAdminOrder qui permet de changer l'affichage de la liste de commande dans le back-office. L'analyse que j'ai faite du code et les différents tests réalisés m'ont montré qu'il suffisait de modifier la requête SQL sélectionnant les informations à afficher et la variable field_list.

Refonte graphique de la partie « Mon Compte »

Pendant que j'intégrais cette fonctionnalité, Léo Biston, l'autre stagiaire qui travaillait avec moi, a créé des maquettes graphiques pour les différentes pages de la partie « Mon Compte ». La suite de mon travail fut donc d'intégrer ces maquettes.



<!--Page de la partie mon compte en taille PC-->

En même temps que j'intégrais, je devais réfléchir à la façon dont ces pages s'afficheront sur différentes tailles d'écran. Le site utilisant le framework Bootstrap, l'utilisation des colonnes et de certaines classes ont été d'une grande aide. Connaître ce framework a été un grand plus dans ce travail, et m'a permis d'être beaucoup plus efficace.

J'ai essayé de rendre les différents éléments le plus lisible possible tout en ne retirant aucune information. J'ai réalisé cette adaptation au format téléphone en navigant sur le site et en me plaçant avec un point

de vue de client. En somme, j'ai appliqué la méthode des scénarios et d'analyse de sites que nous avons pu voir en MMI.



<!--même page en taille téléphone-->

Etant donné qu'il ne s'agissait que d'une refonte graphique, la plupart des éléments existaient déjà. Il m'a suffi de les récupérer pour les remettre en forme selon les maquettes graphiques.

Mise à jour du site topconsommables.com



La seconde mission qui m'a été donné consistait en l'amélioration du site topconsommables.com. Ce site est un site de vente de cartouche d'encre et de toner pour imprimante. Quand j'ai commencé mon travail, le site vendait également des cartouches en sachet. Mais Thomas, le dirigeant de ScaleDev, désirait que les sachets disparaissent de la boutique, et que leur stock fusionne avec le stock de la cartouche équivalente.

Ma problématique était donc la suivante : comment faire en sorte que lors de l'import des produits du fournisseur, de cesser d'importer les produits en sachet, mais que l'on ajoute leur stock à leur équivalent en cartouche.

Analyse de l'import

J'ai donc commencé par essayer de trouver comment relier un sachet et une cartouche. La première et principale ressemblance se situait dans leur référence. Les références des produits de la boutique sont construites en une suite de nombre et de lettre, les 5 premiers caractères correspondant à la série à laquelle le produit appartient. La suite étant le plus souvent des lettres, et correspond à la couleur du produit ainsi que certains caractères pour, par exemple, s'il s'agit d'une cartouche XL.

Les sachets ont pour leur quasi-totalité un « S » à la fin de leur référence. Pour ces produits, on récupère le reste de la référence grâce à la fonction `strstr()` qui permet de récupérer une chaîne de caractère en précisant jusqu'à quelle position de la chaîne on veut aller, et cela correspond dans 80% des cas à l'équivalent cartouche du sachet.

Dans les 20% qui restaient, la plupart étaient des produits dont la référence finissait par « BK » mais dont l'équivalent sachet ne possède pas ces caractères dans sa référence.

Pour ces cas-là, il a fallu comparer les 5 premiers caractères de la référence, la contenance de la cartouche et son prix.

Fusion des stocks sachet et cartouche et import des papiers

Une fois cette analyse terminée, j'ai pu commencer à améliorer le script d'import. Ce script se décompose en plusieurs parties.

Le début permet d'initialiser les variables nécessaires à l'analyse du fichier XML envoyé par le fournisseur, ainsi que les différents objets permettant la gestion du stock et des entrepôts.

Ensuite, on récupère le fichier XML contenant les différents produits. En parcourant ce fichier, on commence par créer les catégories dans lesquelles iront les produits, avec une catégorie mère qui correspond à la marque de l'imprimante, puis une catégorie fille qui correspond au modèle de l'imprimante. Ensuite on vérifie que le produit existe. Si c'est le cas on met à jour le produit sinon on en crée un. Avec les variables initialisées plus tôt, on met à jour les stocks avec une fonction de la classe StockAvailable après avoir ajouté ou supprimé le nombre de produit que l'on voulait de ces stocks.

```
// ajout au stock
$stock_manager->addProduct(
    $produitEnCours->id,
    0,
    $warehouse,
    (int)$article['stock'] - $old_stock_sachet,
    1,
    $produitEnCours->wholesale_price
);
StockAvailable::synchronize($produitEnCours->id);
```

<!--Fonction de mise à jour du stock-->

Enfin on récupère les images fournies par le fournisseur et on les enregistre pour chaque produit.

Il a fallu se placer à deux endroits du script pour pouvoir fusionner les stocks de cartouche et de sachet, lors de la création d'un produit et lors de sa mise à jour.

Etant donné que l'on supprime le produit sachet à la fin du script, il fallait pouvoir enregistrer sa référence dans la ligne du produit cartouche correspondant. De plus, lors de la mise à jour du stock, le fournisseur nous envoie le stock du produit sans qu'il n'y ait eu de fusion. Il fallait donc enregistrer le stock du produit sachet qui a été ajouté au produit cartouche. Pour cela, j'ai créé deux nouvelles colonnes dans la table « ps_product » de la base de données.

Une fois ces problèmes analysés, j'ai créé une fonction de mise à jour des stocks. Elle permet dans un premier temps de mettre à jour le stock des cartouches en prenant en compte les problèmes cités plus haut puis de supprimer le produit sachet après avoir enregistré sa référence et son stock.

Plus tard, on m'a demandé de créer une nouvelle catégorie, et d'importer les différents papiers envoyés. Pour cela, j'ai relu le script depuis le début pour identifier à quels endroits des modifications étaient nécessaires. J'ai donc supprimé les exceptions qui étaient présentes pour les papiers dans la création de catégories et adapté les parties qui nécessitaient un changement.

Les produits se créaient correctement, mais il était impossible de l'ajouter correctement au panier. Après avoir cherché pendant plusieurs heures, Benjamin m'a aidé à comprendre le problème, et on s'est rendu compte que les papiers n'étaient pas correctement ajoutés au stock. Ce bug empêchait la sélection d'un transporteur lors de la commande.

GREENECO - GREENECO	
Stocké	Produit
<input checked="" type="checkbox"/>	Cartouche compatible EPSON T1001 - Couleur : Noir

GREENECO - GREENECO	
Stocké	Produit
<input type="checkbox"/>	Cartouche compatible EPSON T1001 - Couleur : Noir

WEBSSELLER - SIEGE - SAS WEBSSELLER	
Stocké	Produit
<input type="checkbox"/>	Cartouche compatible EPSON T1001 - Couleur : Noir

WEBSSELLER - SIEGE - SAS WEBSSELLER	
Stocké	Produit
<input type="checkbox"/>	Cartouche compatible EPSON T1001 - Couleur : Noir

<!--Comparatif fournisseur coché / décoché-->

Il y avait en fait un script qui avait été créé pour résoudre ce problème pour les autres produits, il a donc juste fallu récupérer le contenu de ce script et de l'intégrer au reste de l'import.

Panier intermédiaire et cross-selling

Prestashop permet d'ajouter dynamiquement les produits au panier et de passer par un panier intermédiaire. Cela fonctionne avec de l'ajout en javascript du produit sélectionné dans un block présent sur la page, puis avec l'envoi en Ajax des produits ajoutés dans ce panier intermédiaire dans le panier final.



<!--panier intermédiaire-->

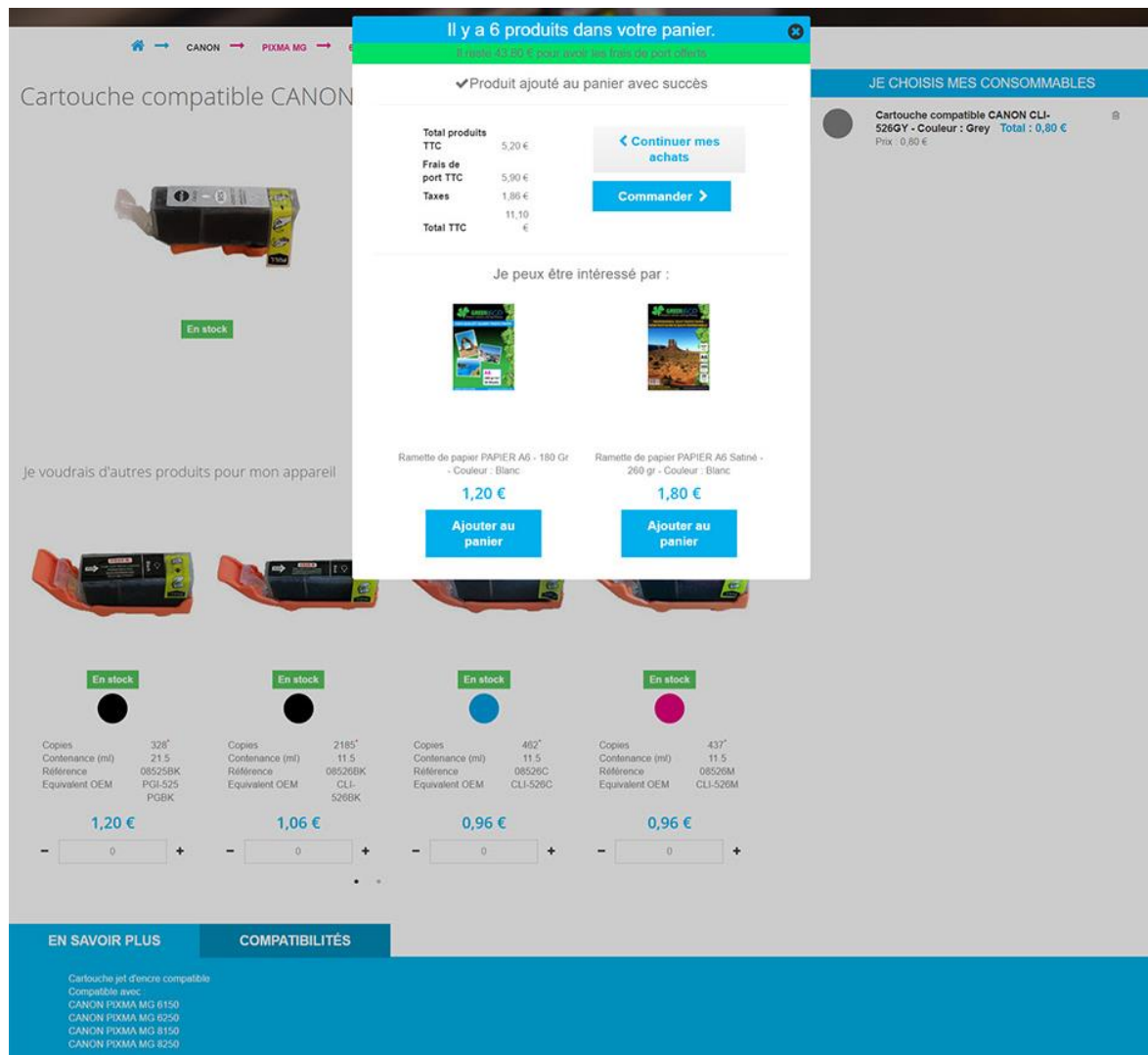
En naviguant et en testant le site après l'avoir modifié, j'ai remarqué qu'il y avait quelques problèmes liés à la modification des pages produits. Par exemple, le nom et la couleur du produit n'apparaissaient pas dans le panier intermédiaire. J'ai supposé que c'était dû à un mauvais nommage des éléments correspondant dans la page.

Grâce à l'inspecteur d'éléments, j'ai rapidement isolé les fonctions JavaScript et les classes HTML qui posaient problème.

Ensuite je devais mettre en place un cross-selling dans le panier intermédiaire. Pour cela, On m'a demandé de créer un formulaire dans le Back-office, qui permet de mettre les id des produits que l'on veut afficher dans le cross-selling. On enregistre ces id dans une ligne de la table « ps_configuration » qui permet d'enregistrer n'importe quelle valeur avec l'identifiant que l'on désire.

Une fois ces id enregistrés, on les récupère dans le controller des produits, et on envoie toutes les données relatives aux produits obtenus dans la vue.

Ensuite, il a suffi de mettre en forme le pop-up d'ajout au panier en récupérant des éléments de la page produit.



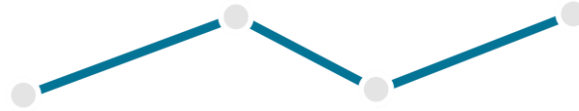
<!--Pop-up d'ajout au panier-->

Un gif lors du clic sur un bouton d'ajout au panier a été rajouté pour montrer à l'utilisateur qu'une action est en cours, car cela n'était pas assez visible au début.

L'ajout au panier se fait comme pour n'importe quel produit, mais on ne passe pas par un panier intermédiaire pour ceux-ci.

Enfin, le panier intermédiaire était sujet à quelques bugs, comme par exemple, lorsqu'on fermait le pop-up d'ajout au panier, le panier intermédiaire ne se vidait pas correctement. Ces bugs étaient liés aux différents fichiers javascript de PrestaShop, que j'ai modifié pour résoudre les problèmes. L'expérience acquise en MMI est un vrai plus dans ce genre de cas, car j'ai pu rapidement identifier les fonctions qui posaient problème et les réparer en conséquence.

Missions secondaires



Au cours de ce stage, les développeurs de la boîte n'ont pas hésité à m'aider lorsque je bloquais sur un problème. J'ai aussi eu l'occasion de les aider sur divers problèmes.

Par exemple, Joris n'arrivait plus à avancer sur l'intégration d'une fonctionnalité de cadeau par tranche de prix sur le site planete-gateau. Il m'a demandé de regarder pour voir si je ne trouvais pas une solution en analysant le problème avec un regard neuf. Suite à cette analyse, on a pu trouver une nouvelle piste de développement, et avec l'aide de Thomas, Joris a finalement réussi à finir l'intégration de cette fonctionnalité.

Une autre fois, Maxime devait récupérer le contenu d'un fichier PDF. Ayant fini mes tâches pour la journée, je l'ai aidé à trouver une bibliothèque qui lui a permis de le faire pendant qu'il développait une autre fonctionnalité.

Ces exemples m'ont montré qu'un regard neuf ou l'aide de quelqu'un d'autre permet d'éviter de perdre inutilement du temps. Le travail dans un open-space permet ce genre d'échange, et facilite le travail du développeur. C'est pour cela que les projets étudiants réalisés à l'IUT sont intéressants, car ils permettent de développer cette mentalité.

Ensuite j'ai aussi eu l'occasion de travailler sur la résolution de bug dans les tunnels de paiement du site de ScaleDev et du site ousoncalin.com.

Dans le cas de ScaleDev, une erreur apparaissait lorsqu'on changeait l'adresse de livraison. En analysant le code, j'ai remarqué qu'une fonction était appelé inutilement à plusieurs endroits du code, et cela provoquait une erreur. En supprimant l'appel de cette fonction, le problème était résolu.

Pour ousoncalin, un client avait fait remonter un problème. Lors de la sélection de la livraison dans un bureau de poste par colissimo pendant la commande, il n'avait pas eu l'occasion de sélectionner le bureau de poste dans lequel il souhaitait être livré. L'adresse qui est remontée dans la commande était au final la sienne.

En essayant de reproduire le même bug, j'ai remarqué que ça n'arrivait que sur le navigateur Safari, lorsqu'on s'inscrivait ou qu'on se connectait dans la page panier. J'en ai déduit que l'erreur se produisait lors du clic sur les boutons pour envoyer les informations de connexion ou d'inscription.

The screenshot displays the checkout interface of ousoncalin.com. It is divided into four main sections: 1. MON PANIER, 2. ADRESSES, 3. MON CHOIX DE LIVRAISON, and 4. MON MODE DE PAIEMENT. In the 'MON CHOIX DE LIVRAISON' section, a table lists various delivery options with their respective costs. The 'BLOC livraison' title is positioned above this table. The table includes options like 'Colissimo Domicile - sans signature' (6.10€ TTC), 'Colissimo Domicile - avec signature' (6.40€ TTC), 'Colissimo - A La Poste' (5.10€ TTC), 'Colissimo - En relais PickUp ou en magasin' (5.10€ TTC), and 'Retrait de la commande à l'atelier' (gratuit). A red box highlights the 'Colissimo - A La Poste' option. Below the table, a note indicates the estimated delivery date: 'Livraison estimée avec ce transporteur entre le Mardi 27 Juin 2017 et le Mercredi 28 Juin 2017'. The page also features a contact number at the top, a cookie consent banner at the bottom, and a 'avis vérifiés' badge showing a 4.7/5 rating.

BLOC livraison	
Choisissez une option de livraison pour l'adresse : Mon adresse	
<input type="radio"/> Colissimo Domicile - sans signature	6.10€ TTC
<input type="radio"/> Colissimo Domicile - avec signature	6.40€ TTC
<input checked="" type="radio"/> Colissimo - A La Poste	5.10€ TTC
<input type="radio"/> Colissimo - En relais PickUp ou en magasin	5.10€ TTC
<input type="radio"/> Retrait de la commande à l'atelier (10440)	gratuit

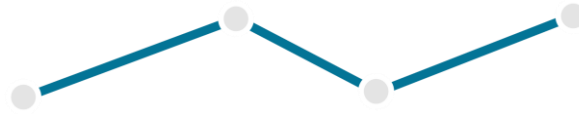
<!--Tunnel de commande ousoncalin-->

Dans les deux cas il a suffi d'appeler une fonction pour recharger les informations du bloc livraison. Cela évite que le pop-up permettant le choix du lieu de livraison n'apparaisse pas.

En réalisant mes tests, j'ai remarqué d'autres bugs. Si on sélectionne la méthode de livraison avant de se connecter ou de s'inscrire, un pop-up s'affiche lors de la sélection du moyen de paiement avec le message « Cadeau ! » et rend impossible la poursuite de la commande.

Pour éviter cela, j'ai choisi de faire disparaître le bloc livraison si l'utilisateur n'est pas connecté et d'afficher un message invitant à s'identifier. Si la connexion est un succès on réaffiche le bloc livraison et on fait disparaître le message.

Conclusion



Mon expérience chez ScaleDev a été très positive pour moi, ce stage étant ma première réelle expérience en développement, en entreprise.

En terme technique, j'ai appris beaucoup de chose, tout d'abord avec l'utilisation du CMS PrestaShop, que je connaissais peu. C'est un outil complexe qui possède une documentation peu fournie. Apprendre à l'utiliser se fait en tâtonnant, mais cela m'a permis d'apprendre à chercher des solutions en analysant le code existant et en parcourant des forums et des documentations.

Grâce aux conseils de Joris, Benjamin, Thomas et Maxime, et à la recherche de solutions aux différents problèmes rencontrés, j'ai pu en apprendre davantage sur PHP et JavaScript et ainsi perfectionner ma logique algorithmique.

J'ai aussi pu me rendre compte de l'intérêt de toujours disposer de papier et d'un crayon. Pouvoir poser ses idées et réflexions sur une feuille est très important et permet d'avoir une première idée du code à produire.

Outre la découverte d'un nouvel outil, je me suis rapidement rendu compte que les méthodes et la logique acquise au cours de ces deux années en DUT MMI sont applicable dans de nombreux cas. La formation offre des bases dans de nombreux domaines qui permettent de réaliser des prestations pour la majorité des clients que nous aurons à rencontrer.

Sur le plan humain et personnel, j'ai pu découvrir comment fonctionnait une équipe de personne ayant la même fonction. Cette expérience m'a permis de comprendre combien il était important de pouvoir travailler à plusieurs sur un même problème, et qu'il valait

mieux perdre 20 minutes à plusieurs, plutôt que de perdre 10 heures sur un problème seul.

Travailler en autonomie a été une très bonne chose pour moi, cela m'a permis d'apprendre et de chercher les choses par moi-même. Cela m'a aussi permis de développer mon esprit d'analyse grâce à la recherche de solutions.

L'ambiance de travail chez ScaleDev est une des choses que j'ai le plus apprécié de mon stage. Le temps de travail est entrecoupé de moment de détente, ce qui permet de se vider la tête et de se replonger plus efficacement dans le travail.

J'ai aussi pu expérimenter des mises en production de mon code. Ce sont des moments assez stressant, mais aussi valorisant. On peut voir que son travail a servi à quelque chose de concret.

Ce stage a été une très bonne expérience, et m'a permis d'être sûr que j'avais choisi la bonne voie avec le développement web. Cela me conforte dans mon choix de continuer mes études en licence MIM, et me donne envie de me lancer dans de nouveaux projets personnels, comme la création d'application mobile, et la découverte de nouvelles technologies.

Glossaire

PrestaShop

Prestashop est une application Web open source (code permettant la possibilité de libre redistribution, d'accès au code source et de création de travaux dérivés) permettant de créer une boutique en ligne dans le but de réaliser du e-commerce.

CMS

Content management system, c'est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites web ou d'application multimédia.

Backend

Terme désignant un étage de sortie d'un logiciel avant de produire un résultat.

Frontend

Terme qui désigne la partie visible de l'iceberg d'un logiciel. Le frontend s'oppose et complète le backend.

SQL

SQL (Structured Query Language) est un langage informatique normalisé servant à exploiter des bases de données relationnelles.

Hook

PrestaShop est un logiciel qui permet de créer des modules ayant pour but d'interagir directement avec l'affichage ou les événements du cœur de la solution.

Les « hooks » ou « crochets » permettent de récupérer ces événements ou encore d'en modifier l'affichage.

Fichier XML

Fichier de données modulaire.

Cross-selling

Une action de cross selling consiste à proposer, au moment de l'acte de vente ou plus tard, la vente d'un produit complémentaire à celui acheté. Elle permet donc l'augmentation du chiffre d'affaires par le biais d'une augmentation du panier moyen.

Ajax

L'architecture Ajax permet de construire des applications et des sites web dynamique interactifs qui communique de manière invisible pour le client avec le serveur.

Résumé

Dans le cadre de mon DUT Métier du Multimédia et de l'Internet (MMI), j'ai effectué un stage de 8 semaines dans une agence web e-commerce, ScaleDev.

Ce rapport va vous permettre de connaître les différentes tâches sur lesquelles j'ai travaillées durant ce stage en développement web, mais aussi de vous rendre compte que ce stage a été pour moi une expérience enrichissante sur le plan professionnel mais aussi personnel.

Abstract

As a part of my formation, DUT Métier du Multimédia et de l'Internet (MMI), I did an 8 weeks internship in an e-commerce web agency, ScaleDev.

This report will describe the different tasks I worked on during this internship in web development, but also express how this internship had been a rewarding experience for me both professionally and personally.