

UNIVERSITÉ DE NANTES

RAPPORT DE STAGE

Développement Web et Solution e-commerce Magento

Aline CHETTA

Master 1 Bio-Informatique et Bio-Statistique

28 mai 2017

Maître de stage :

Thibault VACHER



Tuteur de stage :

Bernard OFFMANN



Remerciements

Je remercie tout d'abord la société Smile pour l'opportunité qu'elle m'a donnée pour effectuer ce stage.

Je tiens à remercier mon maître de stage, Monsieur Thibault VACHER pour m'avoir accueillis dans le service e-commerce.

Je remercie aussi Madame Julie CLABAULT pour avoir pris le temps de m'expliquer l'organisation et les différentes procédures de travail et Monsieur Dorian BRUN pour sa patience et son expertise lors des difficultés rencontrés dans mon travail.

Enfin, je remercie les différentes équipes pour leur bonne humeur et leur accueil.

Table des matières

Remerciements	3
Résumé	7
Abstract	8
Introduction	9
Abréviations	10
Glossaire	11
1 Sujet du Stage	13
1.1 Présentation du Sujet	13
1.2 Présentation des Projets	14
1.3 Objectif du Stage	15
2 Analyse de l'existant	17
2.1 Outils relatifs aux projets	17
2.2 Langages relatifs aux projets	19
2.3 Wacama	19
2.4 Bocage	20
2.5 Speed Burger	21
3 Développement	22
3.1 Organisation du travail	22
3.1.1 Outil de développement	22
3.1.2 Traitement d'un ticket	23
3.1.3 Imputation	24

3.2	Environnement technique	25
3.2.1	Wacama	25
3.2.2	Bocage	26
3.2.3	Speed Burger	27
4	Bilan	29
	Webographie	30

Résumé

J'ai effectué mon stage au sein la société Smile qui m'a permise de participer aux développements web de plusieurs sites e-commerce.

Au cours de ce stage, j'ai intégré plusieurs équipes de développement. Chaque équipe est composé d'un chef de projet et d'un lead developer. Le chef de projet dispatche le travail entre chaque développeur selon leur spécialisation et leur aptitude. Le lead developer a en outre pour tâche de valider ou non les modifications apportées et d'assurer les livraisons du code. Il existe des procédures propres à chaque projet. Si ces dernières ne sont pas respectées, les évolutions apportées ne seront jamais prise en compte. De plus, lorsqu'un développeur contribue à un projet, il se doit de respecter certaines règles de bonnes pratique, les principes de base pour la programmation orientée objet (SOLID) et le dogme " Une fonction fait une seule choses et elle le fait bien ".

J'ai eu l'opportunité de travailler avec la solution Magento 1.x pour les projets Wacama (Camif + Matelsom) et Bocage. J'ai notamment travaillé sur la partie TMA du projet. Ainsi, mes principaux objectifs sont la correction des anomalies, d'assurer rôle de support pour le client et d'intégrer de nouvelles fonctionnalités.

Enfin, le dernier projet auxquelles j'ai participé est Speed Burger. Ce projet est en cours de validation de spécifications ainsi tout reste à faire. Ma principale mission pour ce projet est de développer des modules propres à l'application sous Magento2.

Mots-clé : Magento, Développement Web, Open Source, Git, Redmine, Smile

Abstract

I did my internship at the company Smile which allowed me to participate in the developments of several e-commerce websites.

During this course, I joined several development teams. Each team is composed of a project leader and a lead developer. The project manager dispatches the work between each developer according to their specialization and their aptitudes. The lead developer also has the task to validate or not the modifications made and allow to put code into production.

There are procedures specific to each project. If they are not respected the changes brought will never be taken into account. In addition, when a developer contributes to a project, he must respect a few "good practices" rules, the SOLID's principles and the rule : "A function does one thing and it does good".

I had the opportunity to work with the Magento 1.x framework for the Wacama projects (Camif & Matelsom) and Bocage. I worked on the TMA part of the project. My main focus was to correct anomalies, ensuring support for the client and integrating new functionalities.

Finally, the last project I participated in was Speed Burger. This project is in the process of validating specifications so everything still has to be done. My main mission for this project is to develop modules specific to the application working with Magento2.

Keywords : Magento, Developpement Web, Open Source, Git, Redmine, Smile

Introduction

Dans le cadre de ma formation en Master I Bio-Informatique à l'université de Nantes, un stage de minimum 2 mois est à réaliser afin de valider son année de formation. Ce stage a pour objectif d'acquérir une expérience professionnelle, de mettre en application les connaissances acquises au cours de l'année et de monter en compétence au travers de projets professionnels.

Pour ma part, j'effectue ce stage au sein de la société Smile du 27 Mars au 18 Août 2017. Cette société est connu pour être le premier intégrateur de solution web open source. Ils ont pour clients de nombreuses sociétés françaises et européennes. L'agence de Nantes est composé d'une cinquantaine de collaborateurs et elle est principalement axée sur le développement e-commerce avec la solution Magento et sur le développement web Symfony.

Le sujet de mon stage est de participer au développement web de différents projets e-commerce en cours et ayant pour point commun l'utilisation du langage PHP et de la plateforme Magento. Le but est alors de corriger des anomalies détectées, d'intégrer les nouvelles fonctionnalités et d'assurer un rôle de support pour chaque demande ou problème rencontré par le client.

Afin d'intégrer au plus tôt l'équipe de développement, Smile propose d'effectuer une formation générale au début du stage sur l'environnement de base que rencontre un développeur au sein de la société. De plus, afin de me préparer à mon principale projet de développement (Speed Burger), j'ai également effectué une formation Magento2.

Abréviations

- **CSS** : Cascading Style Sheets : feuilles de styles en cascade utilisées pour définir un ensemble de règles qui régissent l'apparence d'éléments dans une page HTML.
- **HTML** : HyperText Markup Language : format/balise permettant de structurer une page web.
- **TMA** : Tierce Maintenance Applicative : externalisation de la maintenance d'une application d'une entreprise auprès d'un prestataire.
- **XML** : Extensible Markup Language : langage de balise permettant de décrire le contenu d'une page.
- **SASS** : Sassy CSS.POO.MVC

Glossaire

- **Framework** : Ensemble de composants et d'outils formant les fondation de l'application.
- **Merge request** : Requête effectuée au Lead-Developer afin d'appliquer des changements de code d'une branche à une autre. from scratch

Chapitre 1

Sujet du Stage

1.1 Présentation du Sujet

Le principale but de ce stage est de participer à l'élaboration et au maintien d'un site e-commerce. Contrairement aux idées reçues, un site e-commerce n'est pas un site vitrine. Il répond notamment à deux comportements. Le premier est la possibilité pour un consommateur d'effectuer une commande et d'en avoir un suivi. Le second est d'effectuer directement sur le site un achat de produits ou de services par paiement sécurisé. L'atout d'un site e-commerce pour le marchand est de pouvoir accéder à une interface d'administration lui permettant de modifier directement certains paramètres de ces produits.

À la différence d'une application web standard, un site e-commerce doit être en mesure de supporter un important catalogue de produits mais surtout d'être ergonomique, sécurisé et simple d'utilisation pour l'utilisateur. Un site dynamique et correctement référencé est un atout indéniable au succès de l'application. De plus, à l'ère du numérique, les performances d'un site e-commerce doivent être suffisante sur tous les types de surfaces.

Pour se démarquer et assurer le fonctionnement d'un site e-commerce, nombreuses sont les entreprises faisant appel à des prestataires. Smile est

une société connue de l'e-commerce grâce à son partenariat avec Magento. Cette solution open-source se compose du site e-commerce accessible à tous les utilisateurs une fois mis en production et d'une partie back office permettant à l'entreprise de manager ses produits et de connaître le comportement de ses clients par des données statistiques. Le développement sous Magento est réalisé en orienté objet (POO). De plus, il respecte un modèle MVC (Modèle Vue Constructeur) avancée permettant ainsi de classer les dossiers par fonctionnalité et donc de séparer les éléments du front et du back end afin de faciliter le développement. Il dispose également de nombreuses fonctionnalités natives qui peuvent être aisément surcharger pour mettre en place des fonctionnalités sur-mesure demandées par le client¹.

1.2 Présentation des Projets

Actuellement, de nombreux projets sont en cours de réalisations au sein de Smile. La principale mission qui m'a été attribué est de participer au développement du projet Speed Burger. Ce projet consiste à effectuer une refonte complète (from scratch) du site. Néanmoins puisque les spécifications n'étaient pas encore signées à mon arrivé, j'ai également participer aux projets Wacama (Camif et Matelsom) et Bocage.

1. La notion de client désigne l'entreprise pour laquelle la conception d'un site e-commerce est réalisée, cela sera le cas tout au long de ce rapport de stage.

Projets	Principaux Objectifs
Camif	Refonte, Création Api Panier et Tunnel de commande, TMA, SEO
Matelsom	TMA
Bocage	TMA, Mise en place d'une version mobile
Speed Burger	Refonte from scratch, Mise en place pour le lot 1 ² du catalogue de produits, Gestion des flux.

TABLE 1.1 – Tableau récapitulatif des objectifs principaux pour chaque projet auxquelles j'ai participé.

Le tableau ci-dessus montre la grande variété de tâches et d'organisation à effectuer tout le long d'un projet. Chacun des projets est organisé de façon unique (TMA, forfait, régie) selon les différentes mission et selon le cahier des charges établit entre le client et le chef de projet (et l'équipe commerciale).

1.3 Objectif du Stage

L'objectif premier de ce stage est de monter en compétence notamment sur Magento mais aussi sur l'environnement de développement. Pour m'aider, la société Smile m'a permis de réaliser deux formations de 4 jours. La première est la Smile Académie centré sur les outils de développement principalement utilisé au sein de la société et la seconde est portée sur la solution Magento2 et ses bonnes pratiques.

Pour les projets Wacama et Bocage, ma mission est d'assurer une partie des Tierces Maintenances Applicatives (TMA). La maintenance applicative consiste à assurer le bon fonctionnement d'une application, c'est-à-dire de corriger des anomalies, d'adapter le comportement d'un module selon l'utilisation réel d'un consommateur et d'effectuer des améliorations suite à l'évolution de l'application. De plus sur le projet Camif, j'ai pour autre mission

de contribuer au SEO du site web. Pour Speed Burger, elle est de participer principalement au développement des modules sous Magento³ et de respecter les règles de bonne pratiques lors la création des différents modules.

3. Lorsque ce rapport sera rendu, je n'aurai travaillé qu'une semaine sur ce projet.

Chapitre 2

Analyse de l'existant

2.1 Outils relatifs aux projets

Chaque projet est mené sous la direction d'un directeur technique (lead developer). Ce dernier se charge d'initier le projet en créant une lxc permettant l'installation de la solution Magento, de modules tiers, d'éléments de configuration du projet et de mettre en place un environnement semblable à l'hébergeur de l'application. Une lxc n'est pas une machine virtuelle, elle permet uniquement de créer un environnement neutre, non influencé par des distributions Linux. Ainsi lors de la migration en production, aucun conflit n'est à prévoir avec l'environnement du serveur.

Le deuxième outil essentiel est Git. Ce dernier est un logiciel de gestion de versionnage. Chaque projet dispose de son dépôt distant, privé, hébergé par GitLab. Ainsi chaque collaborateur doit être en mesure d'accéder à ce dépôt mais également d'y contribuer pour permettre l'avancement d'un projet mais surtout la collaboration.

Lors de la création d'un tel projet, il est conseillé de construire trois branches. Une branche est une version alternative d'un projet. D'un point de vue simplifié, il s'agit d'un état différent du dépôt mais ayant une même origine (Schéma explicatif). La branche par défaut *master* correspond à l'application présente en production. La seconde est la branche *recette*. C'est au sein

de cette branche que le client peut constater les modifications, effectuer des retours ou valider les changements opérés. La dernière branche est la branche *preprod*. Elle correspond à la branche de développement c'est ici où est validé le travail terminé de chaque développeur par le lead developer. Donc lorsqu'une amélioration ou une correction du code est effectuée sur la branche préprod, elle est déployée sur l'environnement de recette ou de production grâce à l'application Ansible.

Pour dispatcher le travail entre les développeurs, le chef de projet crée des tickets (tâches à effectuer selon les demandes du client) sur Redmine et les attribue aux développeurs du projet. Lorsque un ticket est accompli, il doit être validé par le lead developer puis par le chef de projet. Puis si aucun retour n'est communiqué, il sera inclus lors de la prochaine migration en recette ou en production. De plus, au sein de cette application un wiki est présent. Ce dernier permet d'expliquer la démarche à suivre pour paramétrer certains éléments du projet mais également de connaître les modules présents et d'accéder aux différentes URL du projet (site web / back office).

Enfin, concernant l'architecture des serveurs de production, ils sont généralement organisés de façon suivante.

Nginx -> Varnish -> Apache

Lorsqu'un utilisateur rentre le chemin du site e-commerce, la requête HTTPS est traitée par Nginx. Ce dernier notamment les redirections 30X lors des erreurs de connexion. Puis, il renvoie la requête HTTP à Varnish qui a pour principale fonction de gérer le cache de l'application. Ainsi lorsque l'utilisateur demande une page, Varnish vérifie si les données sont stockées dans son cache. si ce dernier est vrai, il affiche directement la page sinon il

génère la page en demandant à Apache les informations nécessaires.

2.2 Langages relatifs aux projets

La solution Magento est construite à partir du PHP. Par conséquent, l'environnement web nécessite l'utilisation d'une version PHP compatible, du CSS et du JavaScript.

Afin de simplifier et améliorer la visibilité du code, des extensions et des modules complémentaires ont été installés par l'intermédiaire de composer (gestionnaire de paquets libre), notamment Gulp. Ce dernier permet d'automatiser l'ensemble des tâches CSS. Par exemple, il est possible de compiler des fichiers SASS (.SCSS) en CSS. Le SASS contrairement au CSS est un vrai langage de programmation permettant la création de variable et surtout la possibilité d'imbriquer des règles de style. De plus, Gulp est également utilisé pour automatiser l'ensemble des tâches CSS. Concernant le JS, au moins deux frameworks sont jQuery et Prototype présent nativement dans Magento. Comme pour le CSS, un paquet permettant la compilation automatique du Js est mis en place : Grunt.

2.3 Wacama

Ce projet regroupe la mise en place de deux site e-commerce sous Magento gérer jusqu'à présent par un seul back office. Ainsi, ces applications possèdent les composants. Smile collabore avec ce groupe depuis plus de 5 ans. De nombreux modules tiers ont été mis en place. Seul une liste des composants utilisé lors de mon stage est présenté.

Fredhopper Ce service permet d'améliorer le temps de réponse lors d'une recherche de produits. Il est également capable de créer des facettes (critères de sélection) et de trier les produits dans une catégorie.

Akaneo Cette solution est une PIM (Product Information Management). Elle permet la gestion du catalogue de produit en un seul et même point. En autre terme, il s'interface entre chaque systèmes tiers afin de centraliser l'information et ainsi de contrôler la qualité des données et d'exporter ces données aisément entre chaque module(exemple : référence différente des produits ente MySQL et FredHopper).

Mirakl C'est un outil de gestion de marketplace. Il permet aux entreprise d'étendre leur offre de produits en incluant d'autres vendeurs.

2.4 Bocage

Au sein de ce projet où Magento a été mis en place, je n'ai exploité qu'un seul outil supplémentaire. Il s'agit de ngrok. Le but de sa fonctionnalité est de créer un tunnel sécuriser du localhost. Dès lors, il est possible de tester son application en cours de production sur mobile ou d'autoriser l'accès à son localhost à un autre collaborateur.

2.5 Speed Burger

Contrairement aux deux autres projets, le site e-commerce est développé sous Magento2. Cette nouvelle version a subi énormément de changement. Ses nouvelles améliorations permettent de mieux supporter les points d'utilisation du site par les consommateurs, de rendre plus rapide le temps de chargements et de supporter un plus grand nombre de produits. Un thème responsive de base est également proposé. Mais surtout une refonte de l'interface d'administration a été réalisée, elle est désormais moderne et intuitive.

Au niveau du code, Magento2 respecte les normes d'écritures du PSR (PHP Standards Recommendation). De plus, puisque la solution est désormais installée par Composer, l'architecture des dossiers a été modifiée. L'organisation des dossiers au sein d'un module de l'application est plus stricte. Cela permet d'augmenter la lisibilité du code mais surtout de respecter le modèle MVC.

Comme dit précédemment dans ce rapport, le projet vient de débiter et toutes les spécifications ne sont pas encore validées par l'entreprise. Par conséquent, aucun autre module tiers n'a été encore mis en place. Au stade actuel, le but est de mettre en place le catalogue de produits, de gérer les flux entre les ERPs et la caisse et permettre la gestion des produits et des services dans le back office.

Chapitre 3

Développement

3.1 Organisation du travail

3.1.1 Outil de développement

La solution Magento dispose d'une architecture complexe. Afin de faciliter le développement un IDE (Environnement de développement) est utilisé : PhpStorm. Cet éditeur permet d'avoir une vue global sur l'ensemble des fichiers du projet et de visualiser leur type. Il dispose d'une véritable auto-complétion, il est capable de prévenir des erreurs syntaxique en temps réel et il permet une navigation rapide en un clic. Cette dernière fonctionnalité est très utile pour vérifier quelle classe est appelé et ou à quoi correspond telle variable. De plus, il détecte automatique si un dépôt Git est présent et synchronise les modifications en comparant avec l'état précédent de la branche.

Avec cet IDE, il est également possible d'ajouter des modules notamment Magento2. Ce dernier permet notamment d'améliorer l'auto-complétion. De plus, il est nécessaire prendre du temps pour le configurer afin que PhpStorm puisse comprendre le langage XML.

Enfin, un outil indispensable pour le débogage PHP est Xdebug. Ce débogueur permet de simplifier et de faciliter la phase de débogue. Si la configuration est réalisée correctement, cet outil est détecté par PhpStorm et il est

dès lors possible de l'utiliser. Une fois activé, il suffit de mettre un pointeur sur la fonction à déboguer et de recharger la page pour visionner directement sur l'IDE les informations sur l'état des variables au moment *t* et connaître la pile des appels de méthodes. À présent plusieurs fonction sont accessibles notamment le parcours de la méthode pas à pas et l'accès à l'inspecteur qui permet de connaître l'état présent d'une variable ciblée.

3.1.2 Traitement d'un ticket

Pour rappel, le chef de projet attribut un ticket sur Redmine à un développeur. Un ticket est composé d'un identifiant et de l'ordre de mission.

Le premier réflexe à avoir est de changer le statut du ticket sur redmine de *nouveau* à *en cours*.

Il existe deux types de tickets généralement les anomalies (Fix) et les améliorations (Feat) en TMA. Lors d'un travail en TMA, il est nécessaire de créer une branche par ticket. La procédure à suivre est expliqué ci-dessous :

1. Création d'une branche à partir de la branche *preprod* ayant la cette nomenclature :

git checkout -b Type-Identifiant

2. Développement et commit en anglais. Un commit est une commande permettant d'enregistrer son travail. Il doit être sous la forme :

git commit -am "Type #Identifiant : Rapide description des modifications effectuées"

3. Création de la merge request en assignant le Lead developer pour validation.

Lors un développement de module, une branche par développeur est suffisante. Néanmoins, il est toujours nécessaire de suivre les règles de commit.

Dès la merge request effectuée, le statut du ticket sur Redmine doit encore être modifié à *A livrer en recette interne*.

Si l'ensemble de ces règles n'est pas respecté, le ticket ne sera pas validé. Le but de cette procédure est de simplifier le travail du Lead developer mais surtout de garder un historique compréhensible des différentes tâches réalisées sur la branche *preprod* pour ensuite livrer le code validé par les clients.

3.1.3 Imputation

Pour facturer au client le temps de travail de chaque collaborateur d'un projet, deux systèmes de calcul sont mis en place.

- Dans Redmine, le temps en heure de travail doit être renseigné pour chaque ticket. Si un ticket a un temps estimé, il est nécessaire de préciser si il y a eu dépassement ou si dépassement il y aura. Cela permet au chef de projet de prendre des mesures afin de respecter au mieux le cahier des charges.

- Un outil interne *Gescom* permet d'imputer son temps de travail par jour selon le projet travaillé, de renseigner les contributions effectuées.

Il est important de bien renseigner son temps sur Redmine et de prévenir le chef de projet si le temps nécessaire à la tâche est insuffisante car le client y a accès et donc il peut demander des comptes s'il s'aperçoit des anomalies ou des temps de dépassement trop fréquents. Il ne sert à rien d'estimer le

nombre d'heure d'un projet à la baisse pour faire diminuer les coûts.

3.2 Environnement technique

3.2.1 Wacama

Pour apprendre, il n'y a rien de mieux que la pratique. Dès la première semaine, j'ai commencé à traiter des tickets concernant la TMA.

Dans un premiers temps, j'ai travaillé sur des tickets simples consistant à faire des modifications CSS et de corriger des anomalies de premier niveau. J'ai notamment entre autre développé une pop-in permettant l'affichage d'une description pour les frais de traitement dans le panier et dans le tunnel de commande. Comme ce projet concerne deux sites web différents, les demandes d'améliorations ou de correctifs sont généralement réalisées sur les deux. J'ai réalisé ce ticket en début de stage en mettant en place une fonction jQuery et du CSS. Mais, lors de la refonte de Camif et de la création de l'Api Panier, j'ai dû de nouveau retravailler sur ce ticket, d'où l'importance de la TMA. De plus, avec l'expérience gagnée, j'ai pu simplifier le code, notamment la partie SCSS.

De plus, j'ai également travaillé sur des modules spécifiques. J'ai notamment corrigé une bug entraînant une erreur dans le tag commander (permet au client d'avoir des informations lorsqu'un consommateur est en train d'effectuer un achat) et j'ai pu mettre à jour la version de Google Analytics.

Pour faciliter mon intégration au sein de l'équipe, j'ai eu par la suite, pour mission de traiter les demandes SEO (Search Engine Optimization) c'est-à-

dire d'optimiser le référencement et donc gagner en visibilité lors d'une recherche sur un moteur de recherche (exemple : Google). J'ai donc contribué à plusieurs ticket ayant dans ce sens. J'ai mis en place des URLs canonical selon le nombre de facettes (critère) sélectionnées par le client. Une URL canonical permettent d'éviter les contenus dupliqués en déclarant une page dite favorite.

En second lieu, j'ai notamment pris contact directement avec le client pour leur signaler qu'une demande concernant le changement de Doctype pour la version mobile n'était pas intéressante car le Doctype HTML5 déjà mise en place est la dernière version proposée et cette dernière englobe les spécification pour le mobile. De plus, suite à une redondance de ticket concernant un même sujet, j'ai pris le temps d'expliquer au client comment fonctionnait la génération automatique du sitemap de l'application et comment désactiver des catégories de produits inutiles.

Enfin, j'ai modifié l'ensemble du site Camif après la refonte pour que ce dernier soit conforme au standard HTML c'est-à-dire de vérifier et de corriger le balisage pour qu'une seule balise `<h1>` ne soit présente sur les pages web et de mettre en place de façon logique les différentes balises `<h2>` dans le reste des pages. Ses modifications ont été apportées au niveau du code mais surtout au sein des blocs administrable dans le back office. Un bloc est une zone de la page que le client peu modifier à souhait. J'ai créer un exemple d'utilisation pour chacun des blocs présent sur l'application.

3.2.2 Bocage

J'ai pris part à ce projet suite à l'absence du développeur initial. Par conséquent, j'ai du travailler en autonomie et surtout répondre au caractère *urgent* d'un ticket. Le client a requis la mise en place d'une vidéo dans

leur page innovation. Le personnel a tenté d'insérer la video mais cette dernière refusait de se lancer. J'ai donc tout d'abord pris contact avec eux pour connaître leur démarche. Puis avec de la documentation et un brin de patience j'ai pu trouver la cause du problème et le corriger (absence du bloc permettant l'insertion de la vidéo au sein des propriétés XML).

Enfin, j'ai eu des difficulté à contribuer à un ticket demandant la suppression des images des catégories sur mobile et non à la dissimuler avec du simple CSS. Le point technique consistait a développer une fonction capable d'identifier si la requête HTTPS provient d'un mobile. Il existe des solutions open source capable de le faire mais le but est de ne pas surcharger l'application pour une seule utilisation. Donc, j'ai créé une fonction booléenne *detectHttpMobile()* interrogeant la variable `$_SERVER[/.../]` et j'ai modifié le PHTML pour ajouter la condition et le CSS Media Queries pour modifier l'apparence de la catégorie selon la taille de l'écran. Enfin, pour vérifier l'efficacité du code, l'application a été testé sur plusieurs mobiles (Android, iOS) grâce à l'outil ngrok.

3.2.3 Speed Burger

Le projet étant à son début de développement et dépendant de la validation des spécifications, j'ai pour travailler de créer le formulaire de contact. Pour cela, j'ai du surcharger le formulaire de contact par défaut. Pour pouvoir réaliser les mises à jours de Magento2 sans difficulté, il est interdit de modifier les fichiers présents dans le vendor (code source de Magento). De plus, lorsque que l'on met en place un nouveau module, il est essentiel de penser que le client souhaitera un jour modifier tel comportement ainsi il faut optimiser le code que pour des actions précises puissent être modifiable

dans le back office, par exemple les mails de réception du formulaire ou lors de l’affichage d’une liste d’objet.

Chapitre 4

Bilan

L'objectif de ce stage est participer au développement de projet e-commerce et de monter en compétence au niveau des différents langages web mais aussi d'obtenir les bases techniques sous Magento.

Tout au long de mon stage, j'ai essayé d'être la plus autonome possible. Néanmoins, j'ai encore beaucoup de lacune technique notamment sur la solution Magento. Toutefois, lorsque j'ai rencontré des difficultés, j'ai toujours eu une aide de l'équipe ou des conseils. Certes, il est parfois frustrant de recevoir des remarques disant que tel code pouvant être amélioré ou qu'il n'est pas assez sécurisant lorsque l'on réussit enfin à avoir le résultat escompté. Néanmoins, c'est à partir de ses échecs que l'on progresse.

Enfin c'est la première fois où j'ai travaillé en équipe. L'organisation autour d'un projet est compliquée mais la motivation de l'équipe et l'entraide permet à chaque collaborateur de donner le maximum de lui.

Webographie

- **<https://stackoverflow.com/>** : Stack Overflow est un site d'aide en ligne. <https://stackoverflow.com/>
- **<https://redmine-projets.smile.fr>** : Redmine permet au développeur de mettre en place un wiki sur un projet et de prendre connaissance des tickets à traiter. <https://redmine-projets.smile.fr>
- **<https://magento.stackexchange.com/>** : Stack Exchange est un site d'aide en ligne spécialisé Magento. <https://magento.stackexchange.com/>