

Lemon  
interactive

## **RAPPORT DE STAGE**

**DÉVELOPPEUR WEB & WEB MOBILE**

### **LEMON INTERACTIVE**

1 Boulevard de Valmy  
59650 Villeneuve-d'Ascq

<https://www.lemon-interactive.fr>

### **AFPA**

20 Rue du Luxembourg  
59100 Roubaix

<https://www.afpa.fr>

*Période de stage :*

**17/06/2019 au 06/09/2019**

*Période de formation :*

**17/12/2018 au 14/06/2019**

*Tutrice professionnelle :*

**Sarah AHAJJAM**

Chef de projet technique

*Stagiaire :*

**William HATON**

1 815 2164 1

*Formateur :*

**Pierre BRUSSELLE**

DWWM 2019

## **NOTE DE CONFIDENTIALITÉ**

Ce document pouvant comporter des informations internes à l'entreprise **Lemon Interactive** que je juge de mon propre point de vue « sensibles », sa diffusion et sa consultation par d'autres personnes que le jury tout comme sa reproduction est soumise à mon accord préalable ainsi qu'à l'autorisation de l'entreprise.

# REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont de près ou de loin contribué à la réussite de mon stage et de ma formation dans son ensemble.

Je remercie sincèrement Madame Sarah AHJJAM, chef de projet technique et Monsieur Raphaël ROBIL, directeur général de l'agence Lemon Interactive pour m'avoir donné ma chance et pour la confiance qu'ils m'ont accordé tout au long de mon stage.

Je remercie avec force l'ensemble de l'effectif de Lemon Interactive pour leur bienveillance et leur gentillesse à mon égard et pour m'avoir considéré comme n'importe quel membre de l'équipe tout au long de cette période. Je me suis, assez tôt, réellement senti impliqué et faisant parti intégrante de cette équipe.

Mentions spéciales pour : Monsieur Maxime LACHERÉ, chef de projet technique et voisin de bureau, pour sa disponibilité, son aide, ses conseils et sa patience malgré son planning plus que chargé. Monsieur Jeremy BOUHOUR, chef de projet et développeur talentueux qui m'a aidé à me sortir de quelques situations compliquées tout comme Madame Julie CANEVET, développeuse web. Merci aussi à Monsieur Jeremy GRIMONPONT, chef de projet digital et consultant, avec qui j'ai aimé travailler dans la joie et la bonne humeur. Merci aussi à Madame Claire CATTEAU, chef de projet digital et voisine de bureau, pour ses rires et son engouement tout comme Monsieur Vincent CATTOEN, business developer, une bonne âme et un réel atout au sein de cette équipe. Je n'oublie pas les échanges *nerveux* avec Monsieur Antoine MORETTI, trafic manager. Merci aussi aux personnes composant le pôle UI/UX pour leur proximité et pour m'avoir ouvert à de nouvelles réflexions sur la conception des sites web, Mesdames Nedissa BASSE, Aline LANDELLE, Benedetta PASTORE et Marine HACHIN, également directrice associée. Merci aussi à Madame Julie DEMORTIER, Madame Pauline DAUDE, Monsieur Mickaël VAN BIERVLIET, Monsieur Nassim TALBI, Monsieur Nicolas DE LANFRANCHI, Madame Elyse SPECQUE, Monsieur Thomas GOFFART et je pense n'avoir oublié personne, que l'on me pardonne le cas échéant.

Pour la partie pédagogique, je remercie l'AFPA et mon formateur Monsieur Pierre BRUSSELLE pour ses cours, ses conseils et ses ressources partagées. Cette formation n'aurait pas été la même sans mes valeureux collègues, Madame Pauline HERCOUET, Monsieur Benjamin LIGNY et Monsieur Gabriel MISIURNY pour ne citer qu'eux.

Merci à ma famille, mes proches et mes amis pour leur aide et leur soutien indéfectible tout au long de cette période qui s'est étendue sur plusieurs mois.

# SOMMAIRE

NOTE DE CONFIDENTIALITÉ.....	<u>2</u>
REMERCIEMENTS.....	<u>3</u>
SOMMAIRE.....	<u>4</u>
LISTE D'ABRÉVIATIONS.....	<u>5</u>
RÉSUMÉ.....	<u>6</u>
PRÉAMBULE.....	<u>6</u>
INTRODUCTION.....	<u>7</u>
L'ENTREPRISE.....	<u>8</u>
LE PÔLE TECHNIQUE.....	<u>9</u>
L'ENVIRONNEMENT DE TRAVAIL.....	<u>10</u>
MISSIONS.....	<u>12</u>
CLID INDUSTRIE [Front-end + Back-end] .....	<u>13</u>
CASÉO [Front-end] .....	<u>21</u>
APSYS [Back-end] .....	<u>24</u>
TERRASSE2MOBILHOME [Back-end] .....	<u>29</u>
CONCLUSION.....	<u>35</u>

## LISTE D'ABRÉVIATIONS

<b>ACF</b>	Advanced Custom Field (champs personnalisés avancés) [plugin Wordpress]
<b>BO</b>	Back Office (coté administration du site)
<b>CLI</b>	Command Line Interface (Interface en ligne de commande)
<b>CMS</b>	Content Management System (Système de gestion de contenu)
<b>CPTUI</b>	Custom Posts Types (types de contenus personnalisés) [plugin Wordpress]
<b>CRUD</b>	Create Read Update Delete (acronyme) [Créer Lire Modifier Supprimer]
<b>DOM</b>	Document Object Model (Interface de programmation) [Représentation sous forme d'objets d'un document HTML ou XML]
<b>GAR</b>	Abréviation pour signifier Garantie
<b>IDE</b>	Integrated Development Environment (environnement de développement intégré)
<b>MVC</b>	Model View Controller (Modèle-vue-contrôleur) [motif d'architecture logicielle destiné aux interfaces graphiques]
<b>SEA</b>	Seach Engine Advertising (Publicité sur les moteurs de recherche)
<b>SEO</b>	Seach Engine Optimisation (Optimisation pour les moteurs de recherche)
<b>SSH</b>	Secure Shell (Session à distance sécurisée) [programme informatique et protocole de communication sécurisé]
<b>TMA</b>	Tierce maintenance applicative (maintenance appliquée à un logiciel et assurée par un prestataire externe)
<b>UI</b>	User Interface (Interface utilisateur)
<b>UX</b>	User Experience (Expérience utilisateur)

## RÉSUMÉ

Dans le cadre de ma formation de « développeur web & web mobile », j'ai souhaité réaliser mon stage dans une entreprise dont l'activité principale soit à mon sens réellement en adéquation avec ce métier et me suis donc naturellement tourné vers les agences dites de création de sites internet plutôt qu'une société lambda souhaitant créer son site internet à moindre coût.

Ayant déjà un socle de connaissance du métier de par le passé et mon expérience d'autoentrepreneur mais aussi de façon à parfaire mes connaissances auprès d'un effectif compétent, j'ai souhaité également intégrer une agence solidement installée depuis plusieurs années et non une start-up.

J'ai pu intégrer l'agence Lemon Interactive qui se destine aux « moyens et grands » comptes et qui est installée sur la métropole Lilloise depuis plus de dix ans. J'ai pris part à l'activité de l'entreprise comme n'importe quel collaborateur et ai travaillé sur plusieurs comptes clients.

## PRÉAMBULE

Lors de ma recherche de stage, j'avais sélectionné moins de dix entreprises sur la région avec une préférence pour l'une d'entre-elles, la société Lemon Interactive dont je suivais la communication, le site internet, les références depuis plusieurs mois déjà.

N'étant pas au fait des récentes procédures de recrutement, je me suis lancé sans trop d'illusions en répondant à une annonce pour un poste de développeur PHP, envoyant mon curriculum vitae accompagné d'une demande de stage.

Quelques jours après, je fus contacté par téléphone par la chargée de recrutement de l'agence Lemon Interactive, en ce temps la, Madame Émilie COOCHE. Nous avons échangé sur divers points lors de cet entretien téléphonique de plusieurs minutes et suite à cela, il m'a été demandé de réaliser un test technique afin d'évaluer mes compétences. J'ai réalisé ce test en un minimum de temps, étant à ce moment la en formation tous les jours de la semaine, et pas forcément de la meilleure des manières mais j'ai rempli tous les objectifs demandés et le test était pleinement fonctionnel.

Suite à l'envoi de ce test, un rendez-vous dans les locaux de l'agence me fut proposé et je fus reçu par Madame Sarah AHJJAM, chef de projet technique et par Monsieur Raphaël ROBIL, fondateur et directeur général de l'agence. De nombreux échanges et une bonne entente sont ressortis de cet entretien et je fus rappeler en fin de journée et c'est avec une grande joie que j'ai alors été informé de l'acceptation de ma demande de stage.

# INTRODUCTION

Mon premier jour de stage a démarré par un petit déjeuner, coutume de l'agence, qui organise cela pour chaque nouvel arrivant puis j'ai pris possession de mon bureau au sein de l'un des deux open-spaces et me suis attelé à la configuration de mon poste.

Lemon Interactive a fait du CMS Prestashop son outil de prédilection pour ses projets e-commerce et il m'a donc été demandé dans un premier temps de me familiariser avec la dernière version. J'avais, par le passé, déjà réalisé un site pour un de mes clients avec Prestashop mais c'était avec une ancienne version et il fallait donc me remettre un peu à niveau sur les récentes évolutions. J'ai alors conçu un module simple d'inscription à la newsletter tout en m'aidant de la documentation disponible sur le site officiel.



Lemon Interactive est une entreprise où il y a peu de temps mort et pas mal de projets étaient en cours au moment de mon arrivée. On m'a donc mis rapidement et dans un second temps sur la reprise complète d'un site client en cours de réalisation sous Wordpress dont je détaillerai plus loin les tenants et aboutissants. Ce projet fut étalé sur plusieurs semaines et j'ai pu entre temps travaillé sur d'autres projets dont certains seront également définis dans ce rapport.

Côté vie en entreprise, j'ai grandement apprécié le fait qu'on m'est mis à contribution comme n'importe quel développeur de l'agence, ayant mes propres tickets à traités comme tout le monde, devant prendre la responsabilité des mes actions sur les projets, des mises en préproduction, de gérer mes temps, participant aux daily scrums, aux sprints planning et autres réunions internes etc...

Bien plus qu'un simple stage comprenant des missions, ce fut une réelle immersion au sein d'une structure performante, utilisant des processus spécifiques et des outils adéquats et c'est exactement ce que j'étais venu chercher.

# L'ENTREPRISE

Lemon interactive est une agence web créée en 2009, actuellement située à Villeneuve-d'Ascq et qui dispose également d'une antenne sur Paris. Elle accompagne les PME / ETI et Grands Comptes en se positionnant entre les petites structures qui peuvent être compétitives mais ont des difficultés à proposer un réel accompagnement et les grandes agences certes expertes mais proposant des prix disproportionnés et dont le suivi de projets peut s'avérer complexe.

L'agence propose à ses clients un accompagnement concret sur leur transformation digitale à long terme en établissant des solutions pragmatiques et efficaces aux problématiques digitales du moment.

L'effectif comprend plus d'une vingtaine de personnes et peut se targuer d'une expertise bâtie sur plus de dix années de pratique liées au web et comprends de nombreux talents dynamiques et motivés.

Plus qu'une simple agence de création de site internet, son éventail d'actions comprend :

- Stratégie Digitale
- UI/UX
- SEO/SEA [Google Partners certified]
- Site internet e-commerce/institutionnel
- Formations

Quelques clients classés par ordre alphabétique :

APSYS, ANKAMA, BOULANGERIE LOUISE, BUSINESS & HAPPINESS, CAISSE D'ÉPARGNE, CASÉO, CERF DELLIER, CHAUSPORT, CUVELIER FAUVARQUE, DLM, DORIS GROUP, FIDAL, GO INDUSTRIE, HD SOUDAGE, IFPASS, MACSF, NATIXIS, PAUL, POLE SANTÉ TRAVAIL, PAMPLEON, PROJEX, SCOOTE, STARWAX, SUNDIS, TERRES & EAUX, TNP...

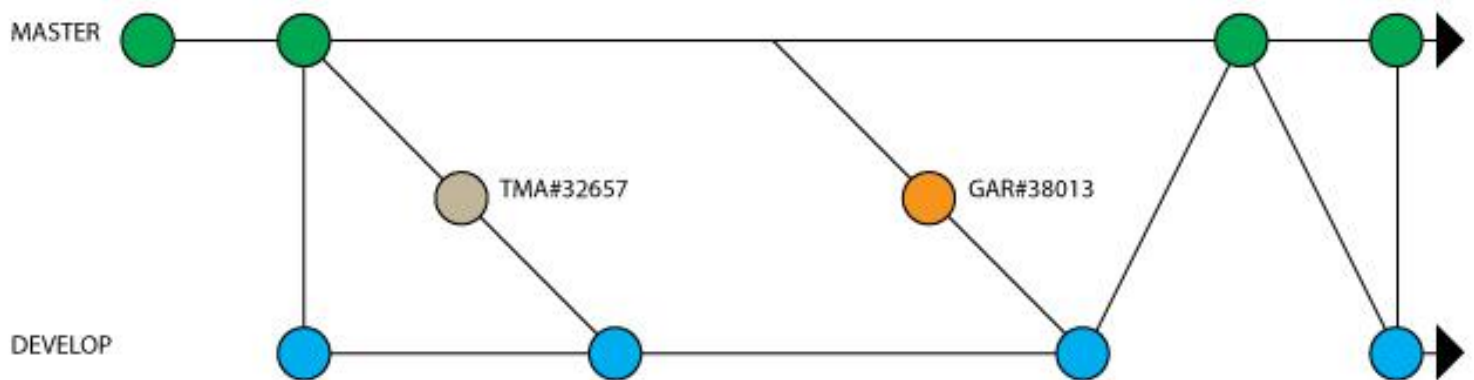


# LE PÔLE TECHNIQUE

Lemon Interactive comprend plusieurs pôles d'activités et en tant que stagiaire développeur, j'ai intégré le pôle technique dont voici sa composition :

- Madame Sarah AHHAJAM, chef de projet technique
- Monsieur Maxime LACHERÉ, chef de projet technique
- Monsieur Jeremy BOUHOURL, chef de projet technique et développeur
- Madame Claire CATTEAU, chef de projet digital
- Monsieur Jeremy GRIMONPONT, chef de projet digital
- Madame Julie CANEVET, développeuse web
- Monsieur Mickaël VAN BIERVLIET, développeur web
- moi même, William HATON, développeur web stagiaire

Les développeurs travaillent sur des projets en créant des branches sur Git et ayant un libellé qui se rapporte au numéro de ticket de la demande. Une fois le travail effectué, les chefs de projet techniques vérifient que le travail est conforme aux attentes, retourne le ticket si besoin ou le valide le cas échéant. Lorsqu'une phase arrive à terme, les chefs de projets mergent les modifications de la branche develop dans la master qui correspond à la branche de production du projet.



Exemple de Gitflow

Une nouvelle branche est toujours constituée à partir de la master. Un libellé lui est attribué correspondant à la demande du ticket. L'exemple ci-dessus montre une branche TMA#32657 qui correspond à de la « tierce maintenance applicative » (un plugin externe a modifié par exemple) et une branche GAR#38013 pour de la garantie et qui peut être une évolution ou un correctif d'un script. Ces branches partent toutes les deux de la master puis sont envoyées dans la branche develop. Si validation, la branche develop est envoyée à son tour dans la master à un temps T et les nouveaux tickets établis après seront donc créés à partir de la master ayant reçus les dernières innovations de la develop.

## L'ENVIRONNEMENT DE TRAVAIL

Les locaux dispose de 2 open-spaces, d'une grande salle de réunion et de deux autres plus petites, d'une salle serveur et d'une salle de repos. Même si il existe des pôles d'activités, Lemon Interactive est une seule et même équipe et a aussi pour coutume de mélanger les personnes au sein des open-spaces. En effet, une fois tous les 3 mois, un nouveau plan d'attribution des bureaux est réalisé et tout le monde change alors de places. Un développeur technique peu donc avoir pour voisin de bureau une personne de l'UI/UX à sa droite et avoir un des deux directeurs associés à sa gauche par exemple.



Un des deux open-spaces

Coté logiciels et services , l'effectif au complet utilise Google agenda pour la planification et mise en commun, Hangouts pour la messagerie interne, OpenERP pour la gestion des heures, Redmine pour la gestion de projets, Owncloud pour le cloud collaboratif.

Pour les sites internet, l'agence a un partenariat avec la société Nexylan qui propose des services d'hébergements infogérés, de monitoring, de Git et autres et dispose d'une infrastructure performante et fiable. Elle est en charge des serveurs de préproduction et de production de la majorité des clients de Lemon Interactive.

OpenERP

Echier Utilisateur Formulaire Options Greffons Raccourcis Aide

Nouveau Enregistrer Supprimer Précédent Suivant Liste Formulaire Calendrier Graphique Imprimer Action Pièces jointes Menu Recharger Fermer

Menu Mes Heures de Travail

Utilisateur : Date : Centre de coûts :

Précédent Suivant

Utilisateur	Date	Centre de coûts	Description	Quantité	Produit	UdM	Montant
William Haton	19/08/2019	Projet Agence / Optimisation de site / Nouvelle	vars email templates		[DEV] Développeur	Heure	-184.75
William Haton	19/08/2019	Client / Client / Création de site / Site Web	logo responsive + breadcrumb + css		[DEV] Développeur	Heure	-184.75
William Haton	19/08/2019	Agence / Bilan	daily scrum		[DEV] Développeur	Heure	-184.75
William Haton	20/08/2019	Agence / Bilan	révision problème js		[DEV] Développeur	Heure	-184.75
William Haton	20/08/2019	Agence / Bilan / Agence	def of done		[DEV] Développeur	Heure	-184.75
William Haton	20/08/2019	Agence / Bilan	daily scrum		[DEV] Développeur	Heure	-184.75
William Haton	21/08/2019	Agence / Bilan	Migrations sites / Migration		[DEV] Développeur	Heure	-184.75
William Haton	21/08/2019	Agence / Bilan	révision problème js / JS		[DEV] Développeur	Heure	-184.75
William Haton	21/08/2019	Agence / Bilan	daily scrum		[DEV] Développeur	Heure	-184.75

La gestion des temps dans OpenERP

TMA #32653

Trouver mon showroom

Ajouté par [William Haton](#) le 19/08/2019. Mis à jour il y a 4 jours.

Statut: Assigné

Priorité: Normal

Assigné à: William Haton

Catégorie: -

Version cible: -

Semaine:

Temps demandé:

Temps assigné:

Demande Team FR: Oui

Début: 20/07/2019

Echéance:

% réalisé: 100%

Temps estimé: heures

Temps passé: -

CPT:

Navigateur:

Environnement:

Open ERP - TMA: TMA

Description

Hello,

Dans le cadre du projet [tma32653](#), nous avons cette optimisation à mettre en place.

Depuis "Trouver mon showroom" dans le header, optimisation du fonctionnement :

- Masquer la liste des magasins au clic sur la zone
- Rajouter une mention dans le champ de recherche "saisissez votre code postal ou n° de département"
- prévoir l'autocomplétion lorsqu'on saisit le code postal
- Si on tape que le n° de département (ex : 59), prévoir d'afficher les résultats (comme c'est le cas sur le formulaire de demande de devis)

Merci

Un exemple de ticket dans Redmine

Projects

Groups

Activity

Milestones

Snippets

T

terrasse2mobillhome

Project

Repository

Files

Commits

Branches

Tags

Contributors

Graph

Compare

Charts

Issues

0

Lemon Interactive

terrasse2mobillhome

Repository

Branches

Overview

Active

Stale

All

Protected branches can be managed in [project settings](#).

Active branches

Y master default

e3fd914e

Merge remote-tracking branch 'origin/tma32417' · 23 hours ago

Y develop

311ad5f3

Merge branch 'tma32417' into develop · 1 week ago

Y tma32417 merged

61c3f8a7

ajout textarea msg devis config · 1 week ago

Y tma32682 merged

13402912

change values control min max devis config · 1 week ago

Y tma32681 merged

Exemple de l'interface Git de Nexylan

11

# MISSIONS

1. **Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.** [Maquettage, interface web statique/dynamique ou travailler sur une solution de gestion de contenu ou e-commerce]
2. **Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.** [Créer/agir sur une base de données, développer les composants d'accès aux données, développer une administration ou travailler sur une administration de solution de gestion de contenu ou e-commerce]

Au cours de ce stage, j'ai couvert tous les objectifs majeurs demandés, ayant travaillé sur plusieurs projets coté front-end et back-end, travaillant des solutions de gestion de contenu et e-commerce, respectant les recommandations de sécurité et portant un grand soucis du détail sur les adaptations responsive des sites qui me furent confiés.

Au cours de mon stage, j'ai pu travaillé sur les comptes clients suivants :

- Apsis [solutions de lavage de contenants industriel]
- Caséo [spécialiste menuiseries et cuisines sur mesure]
- Clid Industrie [solutions en industrie de traitements des surfaces]
- Paul [boulangerie]
- Pôle Santé Travail [actions de préventions des risques]
- Pro racing / PW Minimoto [pièces et équipements mini motos]
- Sports Business Mag [actualités sportives]
- Terrasse2mobilehome [fabricant de bâches sur mesure pour mobile home]

Dans la partie suivante, je détaillerai mes actions sur les projets :

1. **Clid Industrie** [front-end + back-end]
2. **Caséo** [Front-end]
3. **Apsis** [Back-end]
4. **Terrasse2mobilhome** [Back-end]

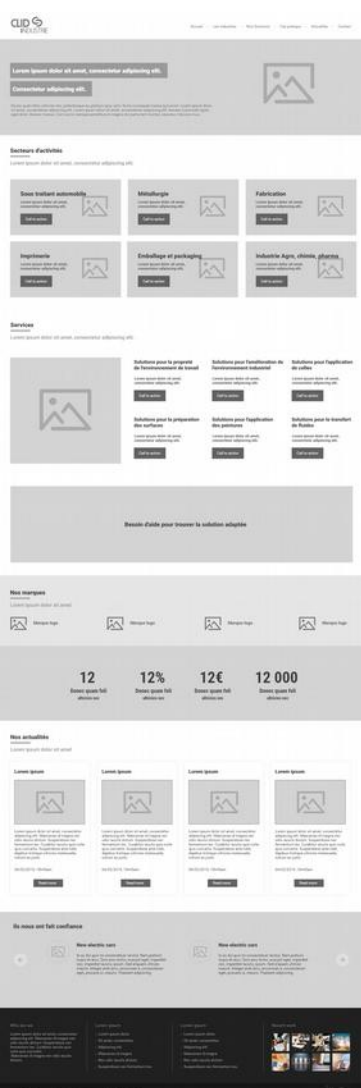
Je dois préciser que le projet Clid Industrie suffirait lui seul à remplir mon rapport de stage tant il couvre un large spectre allant de la conception et création de pages en passant par l'optimisation inpage jusqu'à l'administration complètement repensée et aménagée pour une gestion client.

Je fait néanmoins le choix de découper mon rapport en plusieurs parties, parlant également d'autres projets clients car sur des technologies différentes car c'est aussi plus représentatif de mon stage à savoir travailler sur une multitude de sites ayant des technologies différentes et devoir s'adapter.

Quelques jours après mon arrivée, on m'a proposé de travailler sur un site client en cours de réalisation, 4/5 pages ayant déjà été faites mais nécessitaient une révision suite à l'avancement du projet et des retours clients. Pour ce travail qui s'est étalé sur plusieurs semaines, je fus sous la direction d'un chef de projet, Monsieur Jeremy GRIMONPONT qui m'a orienté, conseillé et avec qui j'ai beaucoup échangé tout au long du processus lors de réunions internes. Le projet fut découpé par lots et un planning d'avancement fut mis en place suivant les validations des spécificités techniques par le client, recettages et autres retours ou nouvelles demandes à prendre en compte.

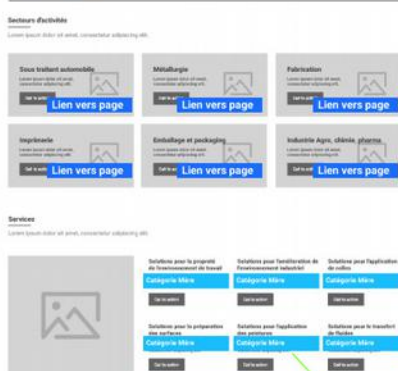
Les besoins : reprise et modifications des pages déjà existantes. Conception des autres pages en fonctions des spécificités techniques et fonctionnelles. Préparation de la partie back-end pour que le client puisse changer ses contenus de lui même. Adaptation du site pour les versions mobiles et tablettes (responsive). Optimisations in-page avant mise en ligne.

On m'a mis à disposition les wireframes et maquettes des pages réalisés avant mon arrivée ainsi que les spécificités techniques et fonctionnelles au format pdf.



## Mise en situation sur les wireframes

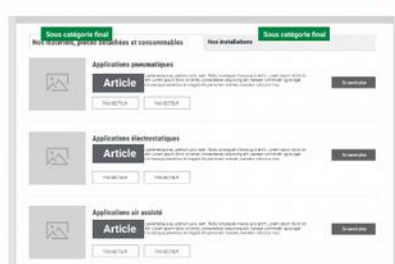
### Home page



### Page industrie



### Page service (Listing catégorie)



Wireframe de la page d'accueil + mise en situation et comportements.



## VISUEL ET TEXTES



- Le visuel de fond, les textes, le bouton (texte et lien) sont administrables depuis le back-office
- Au survol du bouton « EN SAVOIR PLUS » les couleurs sont inversées. (le fond devient blanc et le texte devient vert) le lien dirige l'utilisateur via une encre sur les industries



SPÉCIFICATIONS FONCTIONNELLES PARTIE 1

7

Featured bloc page d'accueil

## MENU MOBILE



- Comportement des sous menus :
  - Au clic le menu glisse pour afficher les liens du sous menu
  - Les liens cliquables à la racine du menu sur la version desktop ne sont plus cliquables mais accessibles depuis le sous menu

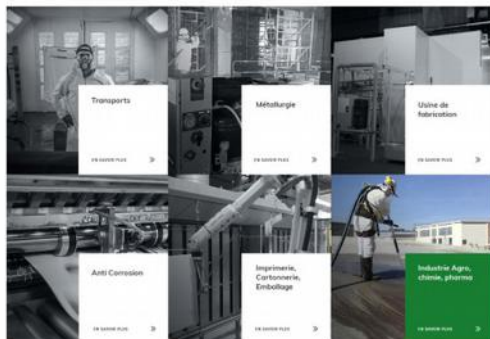


SPÉCIFICATIONS FONCTIONNELLES PARTIE 1

5

Détails pour le menu en version mobile

## ENCARTS DE PUSH DES SECTEURS D'ACTIVITÉS



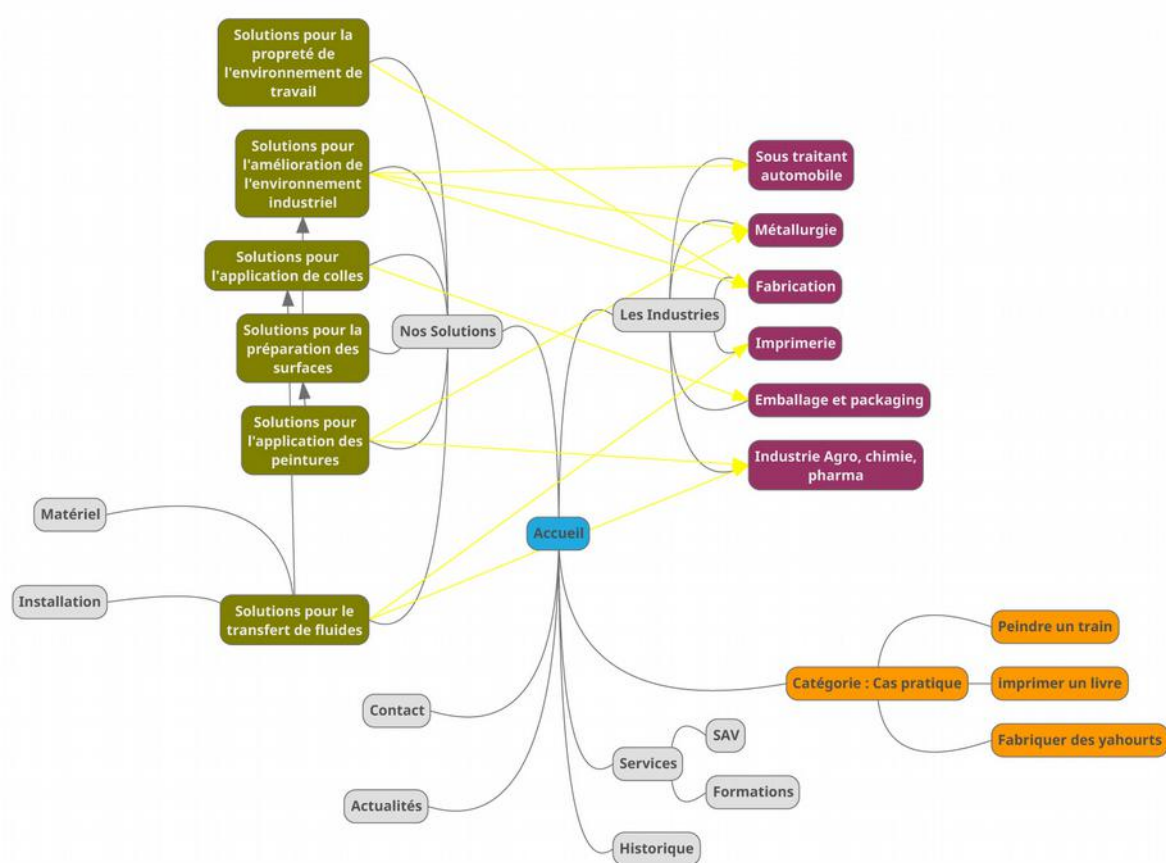
- 6 encarts pour mise en avant des secteurs d'activités
- L'intégralité des informations contenues dans les blocs sont administrables (visuel de fond, titre, lien vers le secteur d'activité)
- L'image d'arrière plan de chacun des blocs devra être uploadée en couleur.
- Taille du visuel 640\*450px
- Au survol d'un bloc, l'image de fond illustrant le secteur d'activité passe automatiquement en couleur, la couleur de fond de l'encart blanc passe en vert, les textes gris passent en blanc.
- Au clic sur un encart, redirection vers la page du secteur d'activité cliqué.



SPÉCIFICATIONS FONCTIONNELLES PARTIE 1

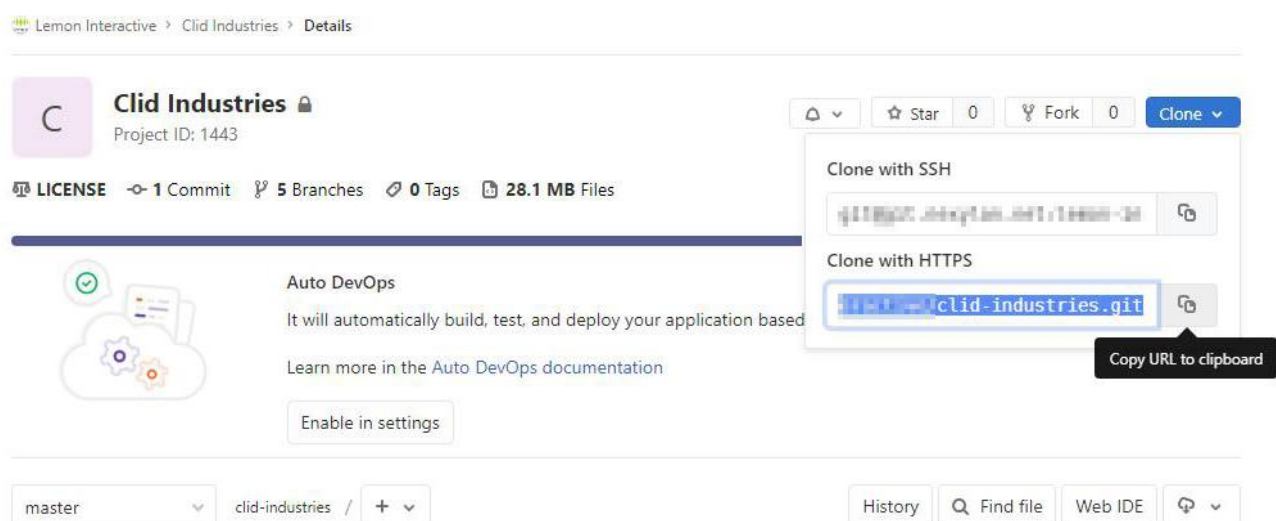
9

Détails pour les encarts des activités

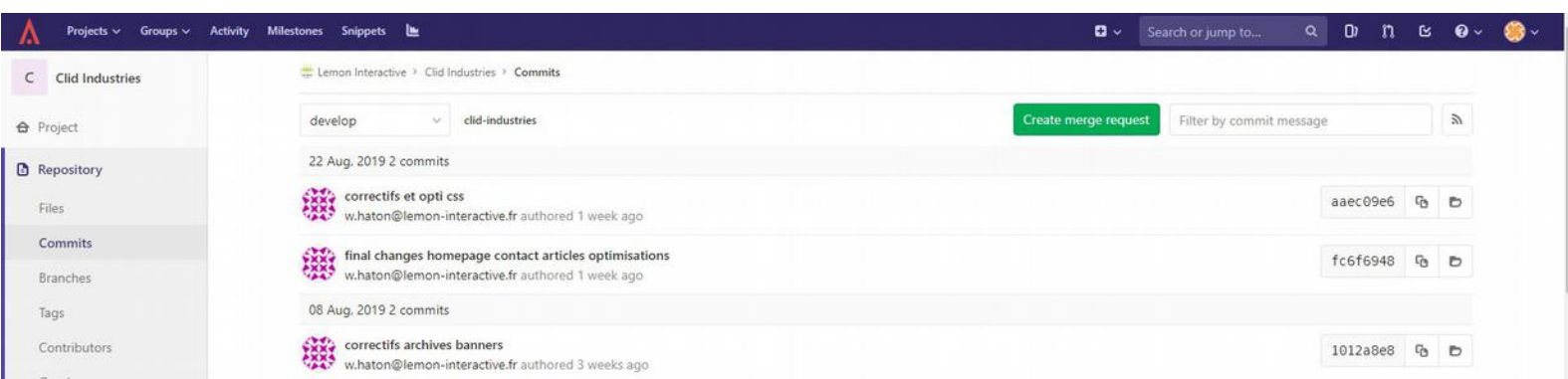


Maillage des liens internes pour les pages

Étant seul désormais à travailler sur ce projet d'un point de vue technique, j'ai cloné le projet sur mon environnement local depuis le Git et j'ai pu travailler directement sur la branche develop. Dès qu'un avancement était terminé, je pushais un commit qui était ensuite vérifié par le chef de projet qui revenait vers moi si besoin ou me donner le feu vert pour passer à la suite.



Interface Git pour le projet Clid Industrie



Commits sur le projet Clid Industrie

Avant d'expliquer certaines de mes actions, je vais tenter de résumer brièvement les fonctionnalités du site. Il s'agissait donc pour le client d'avoir un site administrable et responsive afin d'y présenter ses activités, produits et installations. La structure du site est un Wordpress en version 4 et disposant des plugins ACF, CPTUI et Elementor.

Wordpress comprends nativement 2 types de publications : les articles (posts) et les pages. Les pages sont appropriées pour du contenu fixes, qui changera peu dans le temps comme des mentions légales, un historique ou une présentation. Les articles sont plus utilisés pour une conception de contenu dynamique comme des news qui seront hiérarchisées avec une notion de dates, de classement et d'ordre.

Ces deux types suffisent généralement pour la majorité des sites mais ce n'est pas toujours le cas et c'est pourquoi les plugins CPTUI et ACF furent conçus pour permettent la création de nouveaux types de contenu et de champs avec une gestion distinctes.

Exemples : un site de cuisine a besoin d'avoir des articles de cuisines mais aussi des recettes. Grâce à ces plugins, nous pourront créer en plus du type « article », un nouveau type de contenu appelé « recette » et qui sera indépendant du type « article » mais aura les mêmes fonctionnalités. Un article classique comprend un titre, une description, un corps de texte, un résumé etc... Toujours dans notre exemple de recette, nous avons besoin d'avoir un champs spécifique pour y mettre les ingrédients de notre recette. Avec ACF, nous pourrons créer ce champs en toute simplicité et l'appliquer à la création de chaque nouvelle recette. La combinaison de ces 2 plugins permettent d'étendre considérablement les possibilités de Wordpress.

CPTUI permet de créer de nouveaux types de posts (types articles).

ACF permet de créer de nouveaux champs personnalisés.

Elementor quant à lui est un « builder » de page, permettant une conception de page avec une méthode de glisser/déposer et d'insertion de contenu afin de permettre aux utilisateurs ayant peu de notions de pouvoir construire leurs pages facilement.



Pour ce projet, j'ai donc du créer avec CPTUI les nouveaux types de contenus « activités » pour présenter les secteurs d'activités (métallurgie – fabrication – imprimerie – transport...), « solutions » reprenant les actions de la société pour un besoin (application de peintures liquides – transfert de fluides – équipement des engins de maintenance...), « produits » correspondant à de la fourniture de pièces détachées et consommables (buses – raccords – doseurs ...), « installations » pour des machines et autre structure (cabine – système de dosage – dépoussiéreur...). De nombreux nouveaux champs personnalisés furent créés avec ACF comme : des champs pour des accroches textuelles, des icônes permettant au client de charger des pictogrammes pour la partie « aide » de sa page d'accueil ou des champs bannières permettant de définir une image prenant toute la largeur pour les pages catégories et autres afin de garder une mise en page homogène et affichant un visuel liés au contenu de la page.

Une des spécificités de ce site était de concevoir un système permettant l'affichage d'une image bannière qui prenne toute la largeur accompagnée d'un titre et d'une accroche et incluant une zone de couleur verte par dessus l'image avec un axe spécifique, gestion de l'image et des textes entièrement administrable en BO par le client.

#### Analyse de la demande :

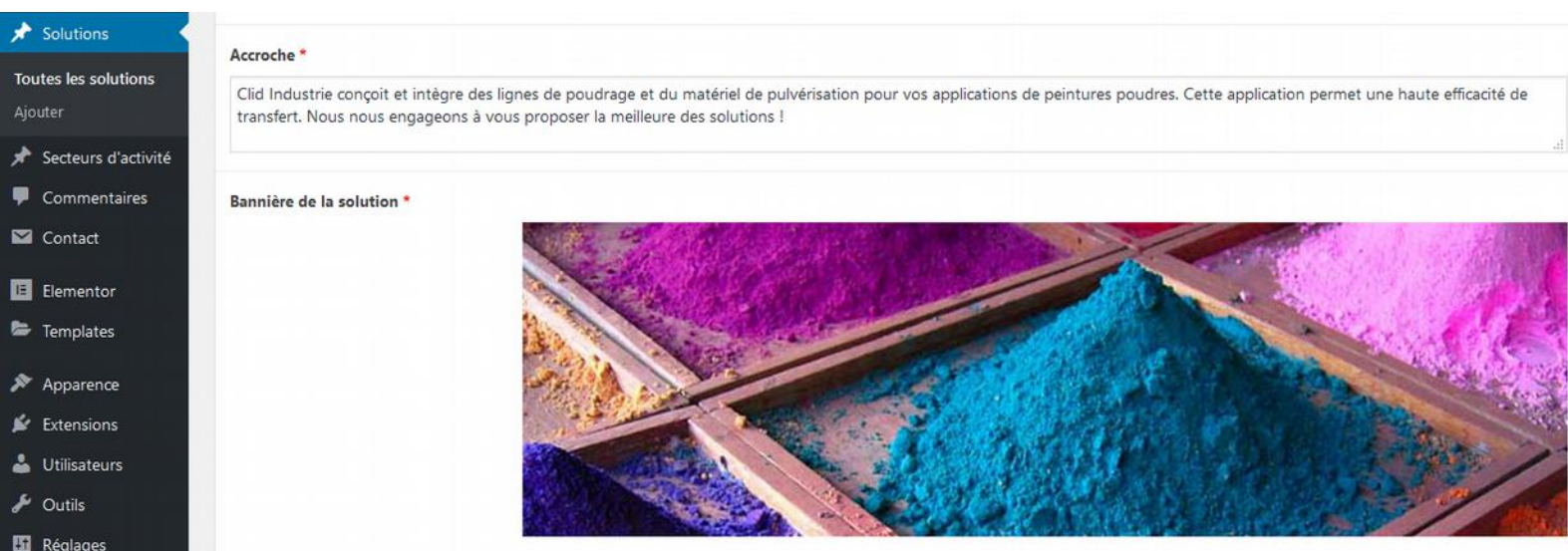
1. Coté administration, besoin de disposé d'un champs pour charger une image spécifique qui servira de bannière. Il faudra aussi un champs titre à cette bannière et un champs pour l'accroche. Ces champs sont affichés lors de la création des pages dont la mise en forme nécessite cet apport comme les pages « solutions » et « industries ».
2. Coté front, besoin d'une zone réservée en haut de page pour la bannière. La zone doit inclure le texte qui servira de titre. L'accroche sera positionné en dessous de la zone bannière. Trouver un système pour la gestion de la partie de couleur verte afin d'éviter de devoir intégré cet aspect à chaque image avant envoi sur le site. Rendre le tout responsive.

#### Moyens définis pour la conception :

Nouveaux champs ACF pour bannière, titre bannière, accroche. La gestion de la zone verte par dessus l'image sera mise en place en CSS. Le réglage de la couverture de la zone sera adapté en fonction de la résolution avec les média-queries en CSS.

#### Conception :

En back-end, dans l'objet « solutions » créé dans ACF, création d'un champs « bannière » de type image et d'un champs « accroche » de type textes multi-lignes avec attribution de ces champs aux pages correspondantes.



Exemple de bannière coté back-end pour la page solution « application de peintures poudre »

Code : dans le template spécifique pour les pages solutions, je défini une variable que j'appelle « \$solutions » et j'utilise la fonction `get_field` de ACF pour récupérer les valeurs de mon objet « solutions ». Cet objet comprends tous les champs créés pour les pages solutions (titre, accroche, image, lien, etc...) Vu que je suis ici dans le template qui sera chargé pour chaque solutions indépendantes, je défini le chargement des bannières au sien d'une boucle « `foreach` » (pour chaque). Le code peut se résumer de la sorte : pour chaque solutions définies, on va aller chercher le champs correspondant à l'image de la bannière. A l'intérieur de mon `foreach`, je défini une nouvelle variable que j'appelle « \$img\_banner » et j'utilise alors la fonction native de Wordpress « `wp_get_attachment_image_src` » pour récupérer l'image attachée au champs `solution_banner` dans la solution en cours d'appel du `foreach`. Le deuxième paramètre correspond lui à une taille d'image définie dans le fichier `function.php` et qui me permet de créer de nouveaux formats d'images. Celui-ci correspond à une image en panoramique pleine largeur. Le résultat de cet appel est un tableau retournant 4 valeurs : l'url de l'image attachée, sa largeur, sa hauteur et un booléen pour définir s'il s'agit d'une taille originale ou pas. La valeur de retour qui nous intéresse ici est l'url. Comme il s'agit de la première valeur d'un tableau, son index est 0 et non 1. `$img_banner[0]` me donne donc l'url de l'image correspondant à la bannière de la solution actuellement chargée par le template solutions.

Afin de créer une zone de couleur verte par dessus l'image de la bannière, je charge donc cette dernière sur la div contenante en passant par CSS en utilisant la propriété `background` et en lui attribuant l'url de l'image. A l'intérieur de cette div contenante, 2 autres divs sont définies. 1 utilisant la class `wrap` possède une largeur maximum et un alignement central afin de garder son contenu au centre de la page. 1 autre div vide qui sera utilisée pour définir la zone verte, toujours en utilisant CSS.

```

1  <?php
2  /*
3  Template Name: Solutions
4  */
5  get_header(); ?>
6  <?php
7  $solutions = get_field('solutions');
8  if ($solutions) : ?>
9  <?php foreach ($solutions as $solution) : ?>
10 <?php $img_banner = wp_get_attachment_image_src($solution->solution_banner, 'page-banner'); ?>
11 <div class="divBannerImg relative hover-transition roboto flex" style="background: url('<?php echo $img_banner[0]; ?>');">
12     <div class="wrap">
13         <div class="contentDivBanner txtShadow">
14             <h1><?php echo $solution->post_title; ?></h1>
15         </div>
16     </div>
17     <div class="angularGreenBkg"></div>
18 </div>
19 <div class="contentSolutionsText">
20     <p class="txtShadow"><?php echo $solution->solution_accroche; ?></p>
21 </div>
22 <?php endforeach; ?>
23 <?php endif; ?>

```

Extrait du template solutions présentant la partie relative aux bannières

La zone verte quant à elle utilise également la propriété background CSS mais elle ne charge pas une image cette fois-ci mais utilise la spécificité « linear-gradient » pour afficher 2 couleurs de fond avec un angle de coupe :

```

.angularGreenBkg {
    background: linear-gradient(110deg, #187f32 50em, transparent 0);
}

```

La première valeur correspond à l'angle, ici 110°, la deuxième est le code couleur vert utilisé, la troisième est une taille pour définir la largeur de la zone en partant de la gauche, la quatrième est pour la seconde couleur sauf qu'ici, j'utilise transparent pour avoir une couleur nulle et la dernière valeur pourrait s'apparenter à l'atténuation. 100 donnera un effet diffus de la transparence, 0 défini une coupe nette et sans effet de transition.

J'ajoute une propriété de text-shadow sur le titre afin que le texte est un effet d'ombrage pour être toujours lisible en cas de bannière claire. Juste pour éviter d'avoir du texte blanc sur un fond clair voir blanc.

## L'Application de peintures poudre

Clid Industrie conçoit et intègre des lignes de poudrage et du matériel de pulvérisation pour vos applications de peintures poudres. Cette application permet une haute efficacité de transfert. Nous nous engageons à vous proposer la meilleure des solutions !

Résultat coté front-end sur un écran d'ordinateur

Pour l'adaptation mobile du système de bannière, il m'a suffit d'utiliser les média-queries afin d'ajuster les valeurs de la propriété background linear-gradient.

```
@media (max-width: 575px) {  
  .angularGreenBkg {  
    background: linear-gradient(110deg, #187f32 16em, transparent 0);  
  }  
}
```

Précision : le thème en place n'étant pas « mobile-first », j'ai dû m'adapter en définissant des média-queries avec l'ancien système de tailles.

On change ici l'étendue de la zone verte en la faisant passer de 50em à 16em pour les écrans compris entre 0 et 575px.

Ci-contre, la page avec la bannière en version responsive smartphone





Caséo est contrairement à l'exemple précédent de Clid Industrie, un site déjà en production. L'équipe continue de travailler dessus et d'apporter des modifications et améliorations en fonction des retours du client. On m'a attribué un ticket concernant la modification du fonctionnement d'un formulaire de localisation de magasins.

Les besoins : le site est un Prestashop et dispose d'un formulaire intitulé « trouver mon showroom » dans la partie header et qui est ce qu'on appelle un « store locator ». Ce formulaire retourne la liste des magasins enregistrés en back-end dans Prestashop. L'affichage actuel liste les magasins par ordre chronologique en fonction du code postal et met donc en avant certains magasins en dépit des autres. Le client souhaite changer ce fonctionnement pour un affichage des magasins par auto complétion tout en gardant plus ou moins la même mise en forme actuelle et celle des champs de formulaires de la partie contact.

Analyse : le site dispose d'autres formulaires notamment dans la partie contact et certains des champs utilisent déjà de l'auto complétion par le biais des plugins JQuery « chosen » et « autocomplete ». Inutile donc pour moi de faire appel à d'autres ressources qui pourraient alourdir le site, je peu tout à fait faire appel à ces mêmes plugins pour cette partie. Le plugin « chosen » transforme une liste en provenance d'une select box classique en un affichage plus moderne en créant un remplacement de cette select box par des divs avec ul et li et peut se couplé avec le plugin « autocomplete ».

Code : le formulaire existant se présente sous la forme d'un module. La liste des magasins est récupérée via une boucle foreach qui va récupérer les informations des adresses magasins créées dans le BO de Prestashop et génère une selectbox coté front.

Le plugin JQuery chosen étant déjà chargé, il ne me reste plus qu'à l'activé sur cette selectbox afin de changer sa structure dans le DOM. Ce plugin comprend plusieurs modes d'actions, permettant les choix multiples et autres options cependant l'utilisation pour notre formulaire est la plus classique et s'active en ajoutant simplement la classe « chosen-select » à notre selectbox. Le plugin va détecté cette classe et agir sur cette selectbox, la cachant en l'entourant d'une div avec un « display none », récupérant ses éléments d'options et générant une nouvelle div par dessous avec une nouvelle structure comprenant un input, une liste (ul li) en fonction du contenu etc...

Tel est le fonctionnement basique de ce plugin. L'autocomplete agit lui sur le input et se réfère également aux résultats des adresses magasins. J'ai ajouté un « append » aux résultats générés afin d'afficher le texte pour « informations, horaires et coordonnées » comme le souhaité le client.



```

themes > interior > modules > belvgstorelocator > belvgstorelocator-nav-content.tpl
37 <div class="storesinfos white-bloc">
38 <div class="formcomplete">
39 <div class="gLabel">
40 {1 s="TROUVER UN MAGASIN : ' mod='belvgstorelocator'}
41 </div>
42 <div class="submitsearch">
43 {if !empty($storesheader)}
44 <form class="formmagazi" action="" method="POST">
45 <div class="item-div">
46 <select class="storesearchheader chosen-select form-control" id="store-select" name="submitstore" onchange="this.form.submit()" data-placeholder="{1 s='Enter your postal code or
47 <option value="">{1 s='Enter your postal code or department number : ' mod='belvgstorelocator'}</option>
48 {foreach from=$storesheader item=store}
49 <option {if isset($defaultstore) && $defaultstore == $store.id_store}selected{/if} value="{ $store.id_store}">
50 { $store.postcode} - { $store.name}
51 </option>
52 {/foreach}
53 </select>
54 </div>
55 </form>
56 {/if}
57 </div>
58 <a class="tousmagasin" href="{ $link->getPagelink('stores')|escape:'html':'UTF-8'}">{1 s='Tous les magasins' mod='belvgstorelocator'}</a>
59 </div>
60 </div>
61 {literal}
62 <script>
63 $(document).ready(function(){
64 // On affiche la liste uniquement à partir de keydown
65 $('chosen-search input').live("keydown",function() {
66 $('chosen-results').css( "display", "block" );
67 });
68 // On recache la liste à keyup
69 $('chosen-search input').focusout("keyup",function() {
70 $('chosen-results').css( "display", "none" );
71 });
72 /*
73 1. On catch l'event showing_dropdown afin de rajouter les éléments à la liste avant la création du chosen
74 */
75 $('chosen-select').on('chosen:showing_dropdown', function(){
76 $('chosen-results > li.active-result + li.active-result').append('<p class="txtOrange">Infos, horaires et coordonnées</p>');
77 }).chosen({width: "100%", no_results_text: "Pas de résultats."});
78 /*
79 2. Régénération de la ligne infos après autocompletion keyup
80 */
81 $('chosen-search input').bind('keyup',function() {
82 if($('chosen-results li:first-child').data("option-array-index") > 0){
83 $('chosen-results > li.active-result').append('<p class="txtOrange">Infos, horaires et coordonnées</p>');
84 } else {
85 $('chosen-results > li.active-result + li.active-result').append('<p class="txtOrange">Infos, horaires et coordonnées</p>');
86 }
87 });

```

Le template front du formulaire store locator

```

<div class="submitsearch">
  <form class="formmagazi" action="" method="POST">
    <div class="item-div">
      <div id="uniform-store-select" class="selector" style="display: none; width: 415px;>
        <span style="width: 399px; user-select: none;>Entrez votre code postal ou département</span>
        <select id="store-select" class="storesearchheader chosen-select form-control" name="submitstore" onchange="this.form.submit()" data-placeholder="Entrez votre code postal ou département" style="display: none;>
          <option value="">Entrez votre code postal ou département
          </select>
      </div>
      <div id="store_select_chosen" class="chosen-container chosen-container-single chosen-with-drop chosen-container-active" style="width: 100%; title="">
        <a class="chosen-single chosen-default">Entrez votre code postal ou département</a>
        <div class="chosen-drop">
          <div class="chosen-search">
            <input type="text" autocomplete="off">
          </div>
          <ul class="chosen-results" style="display: block;>
            <li class="active-result result-selected" style="" data-option-array-index="0">
              ::marker
              Entrez votre code postal ou département
            </li>
            <li class="active-result" style="" data-option-array-index="1">
              ::marker
              03000 - Caséo Moulins
              <p class="txtOrange">Infos, horaires et coordonnées</p>
            </li>
            <li class="active-result" style="" data-option-array-index="2">
              ::marker
              03100 - Caséo Montluçon
              <p class="txtOrange">Infos, horaires et coordonnées</p>
            </li>
          </ul>
        </div>
      </div>
    </div>
  </form>
</div>

```

Le résultat dans la console de l'action des plugins JQuery chosen + autocomplete

Le résultat coté front est visible par étape sur la page suivante.

Étape 1, le bloc est fermé.



**VOTRE MAGASIN**

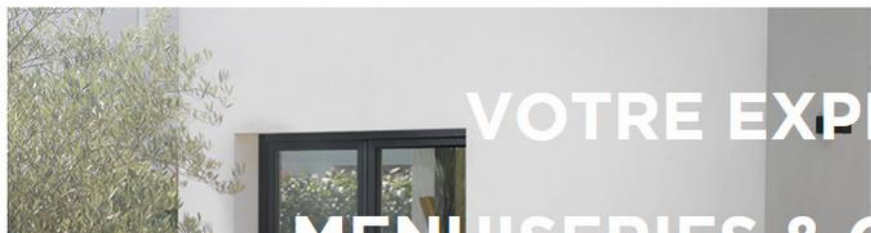
Pour bénéficier des meilleurs offres près de chez vous, choisissez votre magasin de préférence :

**TROUVER UN MAGASIN :**

Entrez votre code postal ou département

Tous les magasins >

Étape 2, le bloc est ouvert et affiche le formulaire à son état initial.



**VOTRE MAGASIN**

Pour bénéficier des meilleurs offres près de chez vous, choisissez votre magasin de préférence :

**TROUVER UN MAGASIN :**

Entrez votre code postal ou département

Étape 3, on accède à l'input pour le démarrage de l'auto complétion.



**VOTRE MAGASIN**

Pour bénéficier des meilleurs offres près de chez vous, choisissez votre magasin de préférence :

**TROUVER UN MAGASIN :**

Entrez votre code postal ou département

- 50110 - Caséo Cherbourg  
Infos, horaires et coordonnées
- 50200 - Caséo Coutances  
Infos, horaires et coordonnées
- 50220 - Caséo Avranches  
Infos, horaires et coordonnées
- 50400 - Caséo Granville  
Infos, horaires et coordonnées

Étape 4, l'auto complétion est effective et la liste se charge en fonction. La sélection d'un magasin fait office de lien direct vers la page concernée.

Apsis, experts en nettoyage industriel, communique par le biais de son site internet réalisé avant mon arrivée par Lemon Interactive. Une des actions du site est de proposer une partie réservée à des vidéos et documents. Cette partie n'est accessible uniquement pour les personnes qui soumettent leur email et acceptent de recevoir la newsletter. La partie restreinte et le module d'accès qui contient le formulaire est déjà en place mais son fonctionnement se contente d'enregistrer les données dans une table et de gérer l'accès à la partie privée or le client souhaite désormais une évolution en back end pour la gestion de ces données.

Les besoins : le site est un Prestashop et le client souhaite accéder aux données des enregistrements en back end du module « blockvideohome » mais aussi pouvoir trier, mettre à jour, ajouter, supprimer et exporter ces données.

Analyse : le site étant déjà en production et ayant déjà des inscrits, j'analyse le fait que je ne dois en aucun cas toucher au fonctionnement actuel afin d'éviter de générer un comportement inattendu mais plutôt trouver un moyen de passer par dessus.

J'opte alors pour la création d'un nouveau module «blockvideomanagement» dont sa fonction sera de lire la table de l'autre module, d'afficher le résultat en utilisant l'interface de Prestashop et de permettre les actions de type CRUD + l'export des données.

Code : je crée mon fichier principal qui définit la classe de mon module et comprend sa fonction « construct » où sont répertoriées les informations du style nom, version, auteur, description, infos de compatibilité et autres...

```
blockvideomanagement.php
modules > blockvideomanagement > blockvideomanagement.php
1  <?php
2  if (!defined('_PS_VERSION_')) {
3      exit;
4  }
5  class blockvideomanagement extends Module
6  {
7      public function __construct()
8      {
9          $this->name = 'blockvideomanagement';
10         $this->tab = 'administration';
11         $this->version = '1.0.0';
12         $this->author = 'WH - Lemon Interactive';
13         $this->bootstrap = true;
14         $this->need_instance = 0;
15         parent::__construct();
16         $this->displayName = $this->l('Admin Module Block Video Home');
17         $this->description = $this->l('Users management for Block Video Home - BO');
18         $this->ps_versions_compliancy = ['min' => '1.7', 'max' => _PS_VERSION_];
19     }
20     public $tabs = array(
21         array(
22             'name' => 'Block Video Management',
23             'class_name' => 'AdminBVH',
24             'visible' => true,
25             'parent_class_name' => 'IMPROVE',
26         )
27     );
28 }
29
```

La classe principale du module



Il me faut une autre classe pour définir les éléments correspondants aux données de la table du module blockvideohome.

La table de ce module ne dispose que de 3 champs :

- un id « id\_videohomeUSER »
- un autre id pour la gestion multi-boutiques « id\_shop »
- un email « email » pour l'adresse email

```
modules > blockvideomangement > classes > BVHManagement.php
1  <?php
2
3  class BVHManagement extends ObjectModel
4  {
5
6      public $id_videohomeUSER;
7      public $id_shop;
8      public $email;
9
10     public static $definition = [
11         'table' => 'videohome',
12         'primary' => 'id_videohomeUSER',
13         'fields' => [
14             'id_shop' => [
15                 'type' => self::TYPE_INT,
16                 'validate' => 'isInt',
17                 'size' => 10,
18                 'required' => true
19             ],
20             'email' => [
21                 'type' => self::TYPE_STRING,
22                 'validate' => 'isEmail',
23                 'size' => 255,
24                 'required' => true
25             ],
26         ],
27     ];
28 }
29
```

La classe des données du module blockvideohome

Le controller admin quant à lui fera la passerelle entre les données du module « blockvideohome » et l'interface UI Prestashop qui fera office de view. J'utilise un `require_once` pour faire appel à ma classe php précédemment créée pour récupérer les définitions des ids et de l'email de la table videohome.

Le construct permet entre autre d'ajuster les options de l'UI de Prestashop et c'est ici que j'autorise l'export des données et les fonctions de tris et de classement par défaut ainsi que les fonction d'update et de détails. Je défini également le formulaire type en cas d'ajout manuel depuis l'interface UI.

```

AdminBVHController.php x
modules > blockvideomanagement > controllers > admin > AdminBVHController.php
1 <?php
2 require_once _PS_MODULE_DIR_ . 'blockvideomanagement/classes/BVHManagement.php';
3
4 class AdminBVHController extends ModuleAdminController
5 {
6     public function __construct()
7     {
8         parent::__construct();
9
10        $this->tab = 'Video Home Admin';
11        $this->bootstrap = true;
12        $this->table = 'videohome';
13        $this->identifier = 'id_videohomeUSER';
14        $this->className = 'BVHManagement';
15        $this->allow_export = true;
16        $this->defaultOrderBy = 'a.id_videohomeUSER';
17        $this->defaultOrderWay = 'DESC';
18        $this->fields_list = [
19            'id_videohomeUSER' => [
20                'title' => 'ID Video Home User',
21                'class' => 'fixed-width-xs'
22            ],
23            'id_shop' => [
24                'title' => 'ID Shop',
25                'class' => 'fixed-width-xs'
26            ],
27            'email' => [
28                'title' => 'Email',
29                'class' => 'fixed-width-xl'
30            ],
31        ];
32        $this->addRowAction('edit');
33        $this->addRowAction('details');
34
35        $this->fields_form = [
36            'legend' => [
37                'title' => 'Video Home',
38                'icon' => 'icon-list-ul'
39            ],
40            'input' => [
41                [
42                    'name' => 'id_shop',
43                    'type' => 'text',
44                    'label' => 'ID Shop',
45                    'class' => 'fixed-width-xs',
46                    'required' => true,
47                ],
48                [
49                    'name' => 'email',
50                    'type' => 'text',
51                    'label' => 'Email',
52                    'class' => 'fixed-width-xl',
53                    'required' => true
54                ],
55            ],
56            'submit' => [
57                'title' => $this->trans('Save', [], 'Admin.Actions'),
58            ]

```

Le controller admin

Je suis les recommandations en matières de traductions qui consistent à écrire les textes au sein de balise prévues pour la traduction et en anglais comme premier langage puis de traduire en français dans un second temps par l'interface prévue à cet effet et qui génère les fichiers de traductions automatiquement.

```

fr.php x
modules > blockvideomanagement > translations > fr.php
1 <?php
2
3 global $_MODULE;
4 $_MODULE = array();
5 $_MODULE['<{blockvideomanagement}prestashop>blockvideomanagement_cf554bdb282d38e4c89bc863b4a0583a'] = 'Admin module pour le bloc vidéo';
6 $_MODULE['<{blockvideomanagement}prestashop>blockvideomanagement_f3f5aff1eeffe8ec2efb2bd4603e3886'] = 'Module d\'administration pour le bloc vidéo en page d\'accueil';
7

```

Le fichier contenant les traductions du module

Le module comme n'importe quel autre est accessible depuis la partie module de Prestashop mais dans un soucis de confort et d'optimisation, je souhaitai qu'il soit directement atteignable par le biais du menu admin dans la colonne de gauche de l'administration du site. Il me fallait pour cela ajouter un bouton à la tab de la partie de mon choix. Je vais ici écrire la partie de la documentation officielle à ce sujet et qui n'est disponible uniquement en anglais à ce jour et en ferait la traduction juste en dessous.

## Anglais :

In order to register new links, open your main module class. We will now use a property called \$tabs, storing an array of link details. Each of them contains a class (= link) to add in the side menu.

### How to define a tab in the menu

Depending on the options you provide, your links won't be displayed the same way:

- **class\_name**: Mandatory, this is the file called when the merchant will click on your link. This is the class name without the Controller part.
- **name**: Optional, this is the name displayed in the menu. If not provided, the class name is shown instead.
- **parent\_class\_name**: Optional if you want to display it in a subcategory. Go farther in this document to see available values.
- **icon**: Optional, will display an icon when the menu is reduced.
- **visible**: Optional, is of boolean type to determine whether you want to display the tab or not.

### How to add names and their translations to a tab

By default, your tab will be displayed in the menu with its class name. If you want to use something more explicit, you can set the name property.

## Français :

Pour enregistrer de nouveaux liens, ouvrez votre classe de module principal. Nous allons maintenant utiliser une propriété appelée \$tabs, en stockant un tableau de détails de lien. Chaque valeur contient une classe (= lien) à ajouter dans le menu latéral.

Comment définir un onglet dans le menu

Selon les options que vous fournissez, vos liens ne seront pas affichés de la même façon :

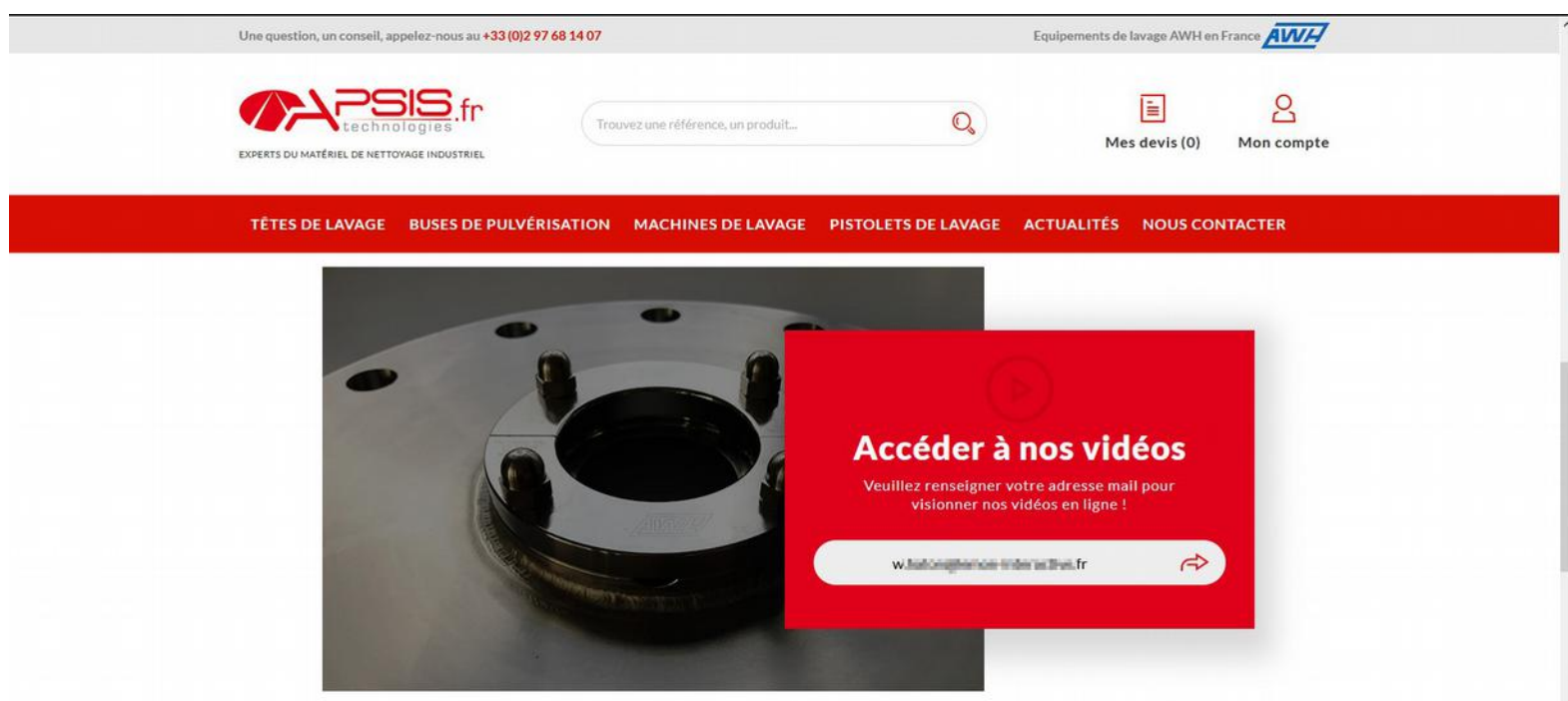
- **class\_name** : Obligatoire, il s'agit du fichier appelé lorsque le commerçant clique sur votre lien. Il s'agit du nom de classe sans la partie Controller.
- **name** : Optionnel, c'est le nom affiché dans le menu. Si non fourni, le nom de la classe est affiché à la place.
- **parent\_class\_name** : Facultatif si vous voulez l'afficher dans une sous-catégorie. Allez plus loin dans ce document pour voir les valeurs disponibles.
- **icon** : Optionnel, affiche une icône lorsque le menu est réduit.
- **visible** : Optionnel, est de type booléen pour déterminer si vous voulez afficher l'onglet ou non.

### Comment ajouter des noms et leurs traductions dans un onglet

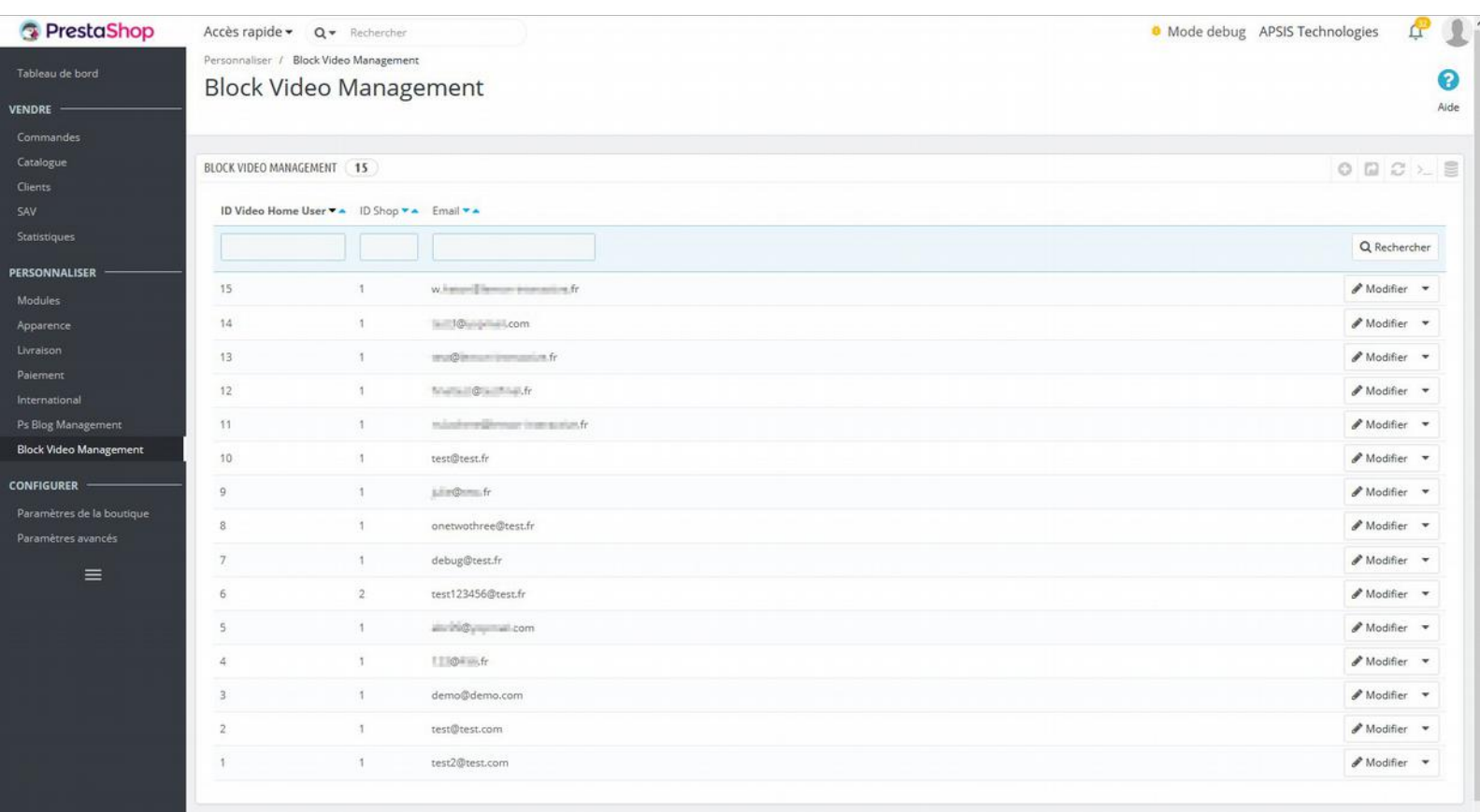
Par défaut, votre onglet sera affiché dans le menu avec son nom de classe. Si vous voulez utiliser quelque chose de plus explicite, vous pouvez définir la propriété name.

Source : <https://devdocs.prestashop.com/1.7/modules/concepts/controllers/admin-controllers/tabs/>

Le résultat est le suivant :



Le module blockvideohome coté front



Le module blockvideomanagement coté back et qui affiche les données du module blockvideohome par le biais de l'interface UI de Prestashop + Bouton « Block Video Management » dans le menu admin



Terrasse2mobilehome est également un site en production et nécessitait quelques évolutions sur sa partie devis. Il s'agit ici d'un site du groupe Nord Bâches permettant la confection de bâches pour terrasses de mobile home par le biais de configurateurs poussés et générant des devis.

Les besoins : le site est un Prestashop et nécessite parmi d'autres modifications, l'ajout d'un champs « message » dans le processus de génération des devis avec configuration afin que le client puisse écrire un message voir une précision pour le traitement de son devis. Le message devra être affichée dans le BO mais aussi dans les emails générés.

Analyse : je repère dans un premier temps les classes, controllers et views qui composent la générations des devis et analyse le code existant afin de me positionner à l'emplacement ou sont définis les éléments du formulaire de devis avec configuration et de quelles façons ils sont utilisés afin de respecter le fonctionnement initial et ne pas perturber le processus par l'ajout de ce champs.

Code : je déclare tout d'abord ma variable \$message au sein de la **classe** de mon devis configuration et lui attribue une visibilité public comme celles déjà en place.

```
Devisconfig.php
classes > Devisconfig.php
1  <?php
2  class Devisconfig extends ObjectModel
3  {
4      public $lastname;
5      public $firstname;
6      public $email;
7      public $phone;
8      public $message;
9      public $fileup;
10     public $validate;
11     public $active;
12     public $date_add;
13     public static $definition = array(
14         'table' => 'devis_config',
15         'primary' => 'id_devis',
16         'multilang_shop' => true,
17         'fields' => array(
18             'lastname' => array('type' => self::TYPE_STRING),
19             'firstname' => array('type' => self::TYPE_STRING),
20             'email' => array('type' => self::TYPE_STRING),
21             'phone' => array('type' => self::TYPE_STRING),
22             'message' => array('type' => self::TYPE_STRING),
23             'fileup' => array('type' => self::TYPE_STRING),
24             'validate' => array('type' => self::TYPE_INT),
25             'active' => array('type' => self::TYPE_INT),
26             'date_add' => array('type' => self::TYPE_DATE)
27         ),
28     );
29 }
```

La classe du devis configurateur



Coté **contrôleur front**, sur la fonction `postProcess` de Prestashop (permet le traitement ajax), je vais récupérer le contenu de la textarea intitulée « message » grâce à la fonction **`Tools::getValue('message')`** de Prestashop et la valeur sera stocké dans la variable `$message`. Je passe ensuite cette variable au sein du devis en cours généré.

```

DevisController.php
controllers > front > DevisController.php > PHP Intelephense > DevisController > postProcess
70  /**
71  public function postProcess()
72  {
73      if (Tools::isSubmit('submitDevis')) {
74          $extension = array('.doc', '.docx', '.pdf', '.png', '.jpg', '.xls', '.xlsx', '.jpeg');
75          $file_attachment = Tools::fileAttachment('fileUpload');
76          $lastname = Tools::getValue('lastname'); // Html entities is not usefull, iscleanHtml check there is no bad html tags.
77          $firstname = Tools::getValue('firstname'); // Html entities is not usefull, iscleanHtml check there is no bad html tags.
78          $tele = Tools::getValue('phone'); // Html entities is not usefull, iscleanHtml check there is no bad html tags.
79          $message = Tools::getValue('message');
80          if (!($from = trim(Tools::getValue('email')))) || !Validate::isEmail($from)
81              $this->errors[] = Tools::displayError('Invalid email address.');
```

Le contrôleur front du devis configurateur

*\* l'appel à la classe « Tools » de Prestashop peut se comparer à l'ouverture d'une boîte à outils, donnant accès à de nombreuses fonctions telles que la génération de redirections, la définition de langues ou de valeurs monétaires et bien d'autres encore. La fonction « `getValue` » et permet de récupérer les valeurs `$_POST` ou `$_GET` d'un formulaire avec les sécurités de Prestashop.*

```

515  /**
516   * Get a value from $_POST / $_GET
517   * if unavailable, take a default value.
518   *
519   * @param string $key Value key
520   * @param mixed $default_value (optional)
521   *
522   * @return mixed Value
523   */
524  public static function getValue($key, $default_value = false)
525  {
526      if (empty($key) || !is_string($key)) {
527          return false;
528      }
529
530      if (getenv('kernel.environment') === 'test' && self::$request instanceof Request) {
531          $value = self::$request->request->get($key, self::$request->query->get($key, $default_value));
532      } else {
533          $value = (isset($_POST[$key]) ? $_POST[$key] : (isset($_GET[$key]) ? $_GET[$key] : $default_value));
534      }
535
536      if (is_string($value)) {
537          return urldecode(preg_replace('/((\\%5C0+)|(\\%00+))/i', '', urlencode($value)));
538      }
539
540      return $value;
541  }
542  
```

La fonction `getValue` de la classe `Tools` de Prestashop

Pour la **view**, coté template front, j'ajoute simplement une textarea au formulaire en place et lui donne « message » pour son attribut « name » afin que sa valeur puisse être récupérée par la fonction Tools getValue citée plus haut. Pour la view, coté template front, j'ajoute simplement une textarea au formulaire en place et lui donne « message » pour son attribut « name » afin que sa valeur puisse être récupérée par la fonction Tools

```

92 {include file="$tpl_dir/errors.tpl"}
93 <form method="POST" id="devisForm" enctype="multipart/form-data" action="{<link>getPagelink('devis', true)|escape:'html':'UTF-8'}">
94   <p class="form-group clearfix">
95     <!-- <label for="lastname">{<1 s='Nom'|sup>*</sup></label> -->
96     <input type="text" name="lastname" id="lastname" value="{if isset($smarty.post.lastname)}{$smarty.post.lastname}{/if}" placeholder="{<1 s='Nom'|sup>*</sup>}">
97   </p>
98   <p class="form-group clearfix">
99     <!-- <label for="firstname">{<1 s='Prénom'|sup>*</sup></label> -->
100    <input type="text" name="firstname" id="firstname" value="{if isset($smarty.post.firstname)}{$smarty.post.firstname}{/if}" placeholder="{<1 s='Prénom'|sup>*</sup>}">
101  </p>
102  <p class="form-group clearfix">
103    <!-- <label for="phone">{<1 s='Numero de téléphone'|sup>*</sup></label> -->
104    <input type="text" name="phone" id="phone" value="{if isset($smarty.post.phone)}{$smarty.post.phone}{/if}" placeholder="{<1 s='Numero de téléphone'|sup>*</sup>}">
105  </p>
106  <p class="form-group clearfix">
107    <!-- <label for="email">{<1 s='Adresse mail'|sup>*</sup></label> -->
108    <input type="text" name="email" id="email" value="{if isset($smarty.post.email)}{$smarty.post.email}{/if}" placeholder="{<1 s='Adresse mail'|sup>*</sup>}">
109  </p>
110  <p class="form-group">
111    <!-- <label for="fileUpload">{<1 s='Télécharger une pièce jointe'|sup>*</sup></label> -->
112    <a class="tips" data-toggle="tooltip" data-original-title="{<1 s='Vous souhaitez nous transmettre plus d'informations ? Des images de votre projet ? Envoyez-nous votre fichier au format .pdf .jpg .png'|sup>*</sup>}">
113      <span>i</span>
114    </a>
115    <input type="hidden" name="MAX_FILE_SIZE" value="{if isset($max_upload_size) && $max_upload_size}{<1 s='Télécharger une pièce jointe'|sup>*</sup>}</input>
116    <input type="file" name="fileUpload" id="fileUpload" class="form-control" placeholder="{<1 s='Télécharger une pièce jointe'|sup>*</sup>}">
117  </p>
118  <p class="form-group clearfix">
119    <!-- <label for="message">{<1 s='Message'|sup>*</sup></label> -->
120    <textarea id="message" name="message" placeholder="{<1 s='Message'|sup>*</sup>}">{if isset($smarty.post.message)}{$smarty.post.message|escape:'html':'UTF-8'|stripslashes}{/if}</textarea>
121  </p>
122  <div class="submit">
123    <button type="submit" name="submitDevis" id="submitDevis" class="button btn btn-default button-medium">
124      <span>{<1 s='Envoyer ma demande et recevoir mon devis par mail'|sup>*</sup></span>
125    </button>
126  </div>
127  <div class="button">
128    <a href="{if isset($urlprecedent) && $urlprecedent != ''}{<1 s='Retour à la configuration du devis'|sup>*</sup>}{/if}" class="button btn btn-default button-medium">
129      <span>{<1 s='Retour à la configuration du devis'|sup>*</sup></span>
130    </a>
131  </div>
132  <div class="content-loader">
133    <div class="loader"></div>
134  </div>
135  <div class="warning">
136    <div class="loader"></div>
137  </div>
138  <div class="content-loader">
139    <div class="loader"></div>
140  </div>
141  <div class="warning">
142    <div class="loader"></div>
143  </div>
144  </p>
145  </div>
146  </div>

```

Le template du formulaire coté front

Il ne manque plus que l'affichage coté back et il me faut pour cela modifié la fonction render du controller admin du devis configurateur afin d'y ajouter le résultat de ma variable « message » ainsi que son view également coté admin pour ajouter une ligne à l'affichage du devis et d'y charger la variable « message »

Copies écrans du controller admin et du view admin sur la page suivante.

```

AdminDeviConfigurateur.php
modules > configurateur > controllers > admin > AdminDeviConfigurateur.php

153
154     public function renderView()
155     {
156         $result = array();
157         $devis = new Devisconfig(Tools::getValue('id_devis'));
158         $result['lastname'] = $devis->lastname;
159         $result['firstname'] = $devis->firstname;
160         $result['message'] = $devis->message;
161         $result['email'] = $devis->email;
162         $result['tele'] = $devis->phone;
163         if ($devis->fileup)
164             $result['file_name'] = _THEME_PROD_PIC_DIR_ . $devis->fileup;
165         $result['date_add'] = $devis->date_add;
166         $result['active'] = $devis->active;
167         $productsList = $devis->getProducts($this->context->language->id);
168         $count = 0;

```

Le controller admin du devis configurateur

```

view.tpl
modules > configurateur > views > templates > admin > dev_configurateur > helpers > view > view.tpl

26 {extends file="helpers/view/view.tpl"}
27 {block name="override_tpl"}
28 <div class="panel">
29     <div class="panel-heading">
30         <i class="icon-comments"></i>
31         {l s="Demande de devis"}
32     </div>
33     <div class="row">
34         <div class="message-item-initial media">
35             <div class="media-body">
36                 <div class="row">
37                     <div class="col-sm-6">
38                         <h2>
39                             {$devis->firstname|escape:'html':'UTF-8'} {$devis->lastname|escape:'html':'UTF-8'}
40                         </h2>
41                         <h2>
42                             {$devis->email|escape:'html':'UTF-8'}
43                         </h2>
44                         <h2>
45                             {$devis->phone|escape:'html':'UTF-8'}
46                         </h2>
47                         <p>
48                             {$devis->message|escape:'html':'UTF-8'}
49                         </p>
50                     <div class="col-sm-6">
51                         <div class="list-action-enable action-enabled">
52                             <i class="icon-check"></i>
53                             <i class="icon-remove hidden"></i>
54                         </div>
55                         <div class="list-action-enable action-disabled">
56                             <i class="icon-check hidden"></i>
57                             <i class="icon-remove"></i>
58                         </div>
59                     </div>
60                 </div>
61                 <div class="message-date">&nbsp;<i class="icon-calendar"></i> - {dateFormat date=$devis->date_add full=0} - <i class="icon-time"></i> {$devis->date_add|substr:11:5}</div>
62                 <div class="message-product">&nbsp;<i class="icon-link"></i> <a href="{ $result.file_name|escape:'html':'UTF-8' }" class="_blank">{l s="Attachment"}</a></div>
63             </div>
64         </div>
65     </div>
66 </div>
67 </div>
68 </div>

```

Le template view admin du devis configurateur

Copies écrans du processus par étapes sur la page suivante.



TOITURE

FAÇADE

JOUE GAUCHE ⓘ

JOUE DROITE ⓘ

TOTAL DU DEVIS  
FRAIS DE PORT INCLUS

138,32€ TTC

## JOUE GAUCHE DE MOBILHOME



Pour compléter la fermeture totale de votre terrasse de mobilhome et créer une pièce à vivre supplémentaire, les bâches de joue sont idéales.....Protection contre la chaleur, contre les vents ou contre les regards indiscrets.

[Prise de mesure d'une Joue Gauche.](#)[Convention de vocabulaire Joue Droite / Joue Gauche.](#)

+ d'infos

PERSONNALISEZ VOTRE JOUE  
GAUCHE

## 1 DIMENSIONS

Type de joue gauche Largeur (cm)\* Hauteur (h) sous poutre de la  
bâche à l'arrière (cm) Hauteur (h) sous poutre de la  
bâche à l'avant (cm) 

## 2 OPTIONS

Coloris\* ☒ Unie ☐ RayéeRails et roulettes ☐ Oui ☒ NonCourroie de relevage ☐ Oui ☒ Non

## 3 LES INFORMATIONS COMPLÉMENTAIRES

Hauteur des fenêtres par  
rapport au sol (cm) 

Fixation sur les poteaux

☐ Oeillets + sandow + poulies attache sandow☐ Oeillets ovales + tourniquets quart de tour

VOTRE JOUE GAUCHE PERSONNALISÉE

138,32€ TTC

Frais de port inclus

\* champs obligatoires

ÉTAPE PRÉCÉDENTE

ÉTAPE SUIVANTE

FINALISER MON DEVIS

DESCRIPTION

PRISE DE MESURE ET DOCUMENTS

SERVICE ET GARANTIES NORD-BÂCHES

La confection des bâches de côtés répond à des **configurations multiples** de votre terrasse de mobil home (rambarde, escalier, porte, la pente du toit...) et des usages variés (bâches sur mesure fixes, coulissantes, relevables....)

Pour chacune de ces bâches de joues, un plan précis doit être établi en fonction des spécificités que vous nous demandez :

Le site présentant un des devis configurateurs en front

TOITURE

FAÇADE

JOUE GAUCHE  
CONFIGURÉE

JOUE DROITE

TOTAL DU DEVIS

138,33€ TTC

Vous avez terminé la configuration de votre bâche.  
Vous serez recontacté pour nous assurer de la conformité des données saisies.

Haton

William

0612345678

w.haton@lemon-interactive.fr

TÉLÉCHARGER UNE PIÈCE JOINTE

Aucun fichier sélectionné

Test message textarea.

ENVOYER MA DEMANDE ET RECEVOIR MON DEVIS PAR MAIL &gt;

RETOUR À LA CONFIGURATION DU DEVIS

Le formulaire coté front avec la textarea

## DEMANDE DE DEVIS

William Haton

0612345678

Test message textarea.

✖ 📅 - 03/09/2019 - ⌚ 09:52

## PRODUITS

Product

Champs pour le calcul

Champs pour informations

Total



Joue gauche de mobilhome

Type de joue gauche : Sans fenêtre - sans porte  
Largeur (cm) : 200  
Hauteur (h) sous poutre de la bâche à l'arrière (cm) \* : 225  
Hauteur (h) sous poutre de la bâche à l'avant (cm) \* : 255  
Coloris : Unie  
Couleur : VERT  
Nombre de Portes supplémentaires :  
Nombre de Fenêtres supplémentaires :  
Volets déroulant :  
Nombre de volets déroulants :  
Fenêtre ouvrance :  
Nombre de fenêtres ouvrantes :  
Rails et roulettes : Non  
Courroie de relevage : Non

Hauteur des fenêtres par rapport au sol (cm) : 188  
Fixation sur les poteaux :

98,33€

Total Produits :	98,325
Frais de port :	40€
<b>Total</b>	<b>138,33€</b>

Le résultat coté admin dans la vue d'un devis : le message est consultable

## CONCLUSION

A travers mon expérience de stage au sein de l'agence **Lemon Interactive**, j'ai découvert le fonctionnement complet d'une réelle agence entièrement dédiée au web, à la gestion des projets et l'accompagnement de moyens et grands comptes.

Mes activités ont été très variées et intenses et m'ont permis d'en apprendre d'avantage et de parfaire mes connaissances et pour finir, de renforcer mon socle de compétences précédemment acquises.

Techniquement parlant, j'ai grandement évolué en PHP/MVC et sur le CMS Prestashop. J'ai eu aussi ce que j'appelle des déclics sur l'utilisation de Git et notamment ma compréhension des dépôts indépendamment des autres et des têtes de lectures (head). Il m'a fallu aussi m'adapter souvent aux changements rapides de technologies, travaillant un jour sur du Prestashop 1.6, le lendemain sur du Wordpress 4.9, le surlendemain, sur du Prestashop 1.7 etc... C'est une gymnastique du cerveau assez intéressante...

J'avoue m'être aussi posé certaines questions durant mon stage, me demandant si je pourrais m'adapter à long terme à une charge de travail conséquente ou si il me serait possible de concilier le temps de rendement et le temps nécessaires à la réalisation d'un travail minutieux. Ces interrogations se sont estompées au fil du temps et je me suis surpris moi même sur ma capacité à réaliser certains travaux dans un laps de temps donné.

A la fin de mon stage, je me suis rappelé de mes années en autoentrepreneur et de pourquoi j'avais entamé ce processus de formation. En effet, travaillant seul et étant focalisé sur mes projets, j'en avais oublié de suivre les évolutions des technologies et n'avait pas perçu immédiatement les changements initiés par les montées en puissance de l'UI/UX et autres adaptations du SEO dans la conception des sites web.

Ma formation m'a permis, en plus d'apprendre et me remettre à niveau, de retrouver de l'échange et de l'entraide avec mes camarades de promotions. Mon stage au sein d'une équipe de plus de 20 personnes a définitivement confirmé mon souhait de continuer à travailler en équipe.

Je sais aussi qu'on apprend constamment et que rien n'est acquis, surtout dans le digital, nécessitant une constante remise en question mais c'est aussi ce qui me plaît et je sais que je peu désormais compter sur des amis sincères et une équipe solidaire les uns des autres.

Cette expérience fut un réel atout pour ma vie professionnelle et même si ce ne fut pas simple et facile tous les jours de par le rythme intense, je suis très content et satisfait de mon parcours et je compte désormais les jours qui me sépare de mon retour à l'agence **Lemon Interactive** pour de nouvelles aventures avec mon équipe !



William HATON  
[DWWM 2019]  
Droits de reproduction  
et de diffusion réservés