



Le génie pour l'industrie

RAPPORT TECHNIQUE  
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
DANS LE CADRE DU COURS GTI/LOG792 PROJET DE FIN D'ÉTUDES EN GÉNIE  
LOGICIEL ET DES TI

**CONCEPTION ET MISE EN PLACE D'UN SITE WEB**

GABRIEL PATEAU  
CHRISTIAN BAMATEMBERA  
PATG13109007  
BAMN05119208  
DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

**Professeur-superviseur  
ALAIN APRIL**

MONTRÉAL, 19 AVRIL 2016  
HIVER 2016

## REMERCIEMENTS

On tient à remercier le professeur Alain April pour ses conseils tout le long du projet, à Maxime Dupré Corriveau pour le temps supplémentaire qu'il nous a accordé pour la mise en place de notre environnement de travail, et de la formation reçu quant à l'utilisation de Wordpress. On tient à remercier aussi monsieur Dan Novosad qui nous a permis de réaliser ce projet.

# Table des matières

Introduction .....	5
Conception de l'architecture du plug-in .....	6
Mise en situation .....	6
Utilisation du patron de conception DAO .....	6
Description du patron.....	6
Avantages et Inconvénients.....	6
Intéractions entre les différentes classes du patron.....	7
Conception de l'interface utilisateur.....	8
Mise en situation .....	8
Besoins .....	8
Prototype.....	9
Ajouter un produit .....	10
Supprimer un produit .....	11
Modifier un produit .....	12
Discussion.....	13
Implémentation du plug-in .....	15
Mise en situation .....	15
Logiciels utilisés .....	15
Contrôle de version.....	15
Environnement de développement .....	16
Gestionnaire de contenu.....	16
Gabarits utilisés .....	17
Gabarit visuel.....	17
Gabarit du plug-in .....	17
Librairies utilisées .....	17

jQuery .....	17
Bootstrap CSS .....	18
SweetAlert .....	18
Formation sur la gestion du site web .....	18
Mise en situation .....	18
Formation sur l'ensemble des fonctionnalités du site .....	18
Assistance au client .....	18
Tutoriel .....	19
Formation sur le plug-in .....	19
Tutoriel .....	19
Ajustement du contenu du site web .....	20
Mise en situation .....	20
Ajout du contenu .....	20
Ajout des fichiers PDF .....	20
Ajout des témoignages .....	20
Modification du contenu existant .....	21
Modification des images du carrousel .....	21
Déploiement .....	22
Mise en situation .....	22
Procédures .....	22
Conclusion .....	23
Recommandations .....	24
Bibliographie .....	26

## Liste des figures

Figure 1 — Diagramme du patron DAO .....	7
Figure 2 — Vue d'ajout de produit.....	11
Figure 3 — Vue de tous les produits ajoutés.....	12
Figure 4 — Vue de modification d'un produit .....	13
Figure 5 — Prototype initial .....	14

## Introduction

Présentement, le client Dan Novosad promeut et vend ses produits en faisant du bouche-à-oreille. Par conséquent, ses produits ne se font pas connaître autant qu'il le voudrait. Avec le commerce électronique qui devient de plus en plus présent sur Internet, le client doit se mettre à-jour et suivre la tendance. Tout d'abord, il faut changer la façon dont il promeut ses produits afin qu'il ait une meilleure visibilité sur Internet. Ensuite il faut que le client garde son autonomie quant à la promotion de ses produits. Puisque notre client est sous-représentant de "Milwaukee valve", il est important que le site puisse offrir le service adéquat afin de rediriger les clients potentiels à Dan lorsque ceux-ci sont à la recherche d'une valve.

Les objectifs du projet étaient, en premier lieu, de créer un site web permettant à Dan Novosad d'afficher de l'information sur les valves qu'il vend, de faire connaître son entreprise et de recueillir les commentaires d'utilisateurs afin d'apporter des améliorations sur les services qu'il offre. De plus, la gestion du contenu du site web doit être faite en grande partie par Dan Novosad puisqu'il veut le gérer lui-même et apporter des modifications au besoin. Le site web était déjà fonctionnel, suite au PFE de l'étudiant précédent, mais le contenu de ce dernier n'était pas à jour. Les tâches restantes à faire dans le projet étaient d'accompagner Dan Novosad à la gestion du site web, adapter le contenu avec ceux qu'il a choisis. Un plug-in servant au visionnement des valves sur le site web afin d'informer les consommateurs a été développé pour permettre à Dan d'informer les clients potentiels des produits qu'il promeut. Parmi les fonctionnalités comprises, il y a l'ajout d'un nouveau produit, le retrait d'un produit, la modification d'un produit et d'autres fonctionnalités permettant d'améliorer l'expérience utilisateur du plug-in

# Conception de l'architecture du plug-in

## Mise en situation

Pour la partie de la conception du plug-in, Gabriel a dû effectuer des recherches sur les différents patrons disponibles et sélectionner ceux qui étaient les plus simples et les plus efficaces à utiliser dans le contexte actuel. Parmi ceux qui ont été retenues, le patron MVC et DAO ont été utilisés pour l'architecture de ce plug-in. L'architecture en question est décrite plus en détails ci-dessous.

## Utilisation du patron de conception DAO

### Description du patron

Le patron de conception DAO (Data access object) est un patron qui permet d'offrir des opérations liées à la récupération de données, suppression de données ou autres opérations demandant une interaction avec la base de données en les encapsulant dans une instance dans laquelle ces opérations y seront uniquement accessibles. En l'utilisant avec le patron MVC (Modèle-Vue-Contrôleur) dans lequel la vue représente l'instance qui envoie ses informations au contrôleur, le contrôleur qui gère les données reçues de la vue et le modèle qui détient les opérations permettant de communiquer avec la base de données, on sera capable de concevoir une architecture qui sera simple et efficace. Pour ce qui est du patron *Data access object*, celui-ci se retrouve entre les contrôleurs et les modèles et sert principalement à séparer la logique d'affaire servant à communiquer avec la base de données de la logique d'affaire servant à gérer les vues. En utilisant ces deux patrons lors de la conception du plug-in, nous avons pu répondre aux besoins de ce dernier.

### Avantages et Inconvénients

Le principal avantage du patron *Data access object*, est qu'il permet de cacher les requêtes faites à la base de données par le biais d'opérations qui seront accessibles par le *Data access object*. Le terme *Data access object* qui porte le même nom que

le patron, est une entité qui permet d'encapsuler les différentes opérations servant à interagir avec la base de données.

Ces opérations peuvent servir par exemple à ajouter, afficher, modifier ou supprimer des informations dans la base de données. Le fait d'utiliser ce patron permet aux vues et aux opérations permettant d'interagir avec la base de données de ne pas se connaître entre eux.

Cette caractéristique s'avère utile dans les cas où on voudrait apporter des modifications au plug-in sans affecter d'autres composants faisant partie de l'architecture ou encore effectuer de la maintenance sur le plug-in.

Par contre l'inconvénient est qu'on multiplie les appels avec la base de données, c'est-à-dire que l'utilisateur est forcé d'utiliser les fonctions provenant du *Data access object* pour accéder à l'information voulue. Parfois, ces fonctions peuvent faire plusieurs appels à la base de données avant qu'ils puissent retourner le résultat escompté, tandis que le résultat pourrait être obtenu grâce à un appel à la base de données.

### Intéractions entre les différentes classes du patron

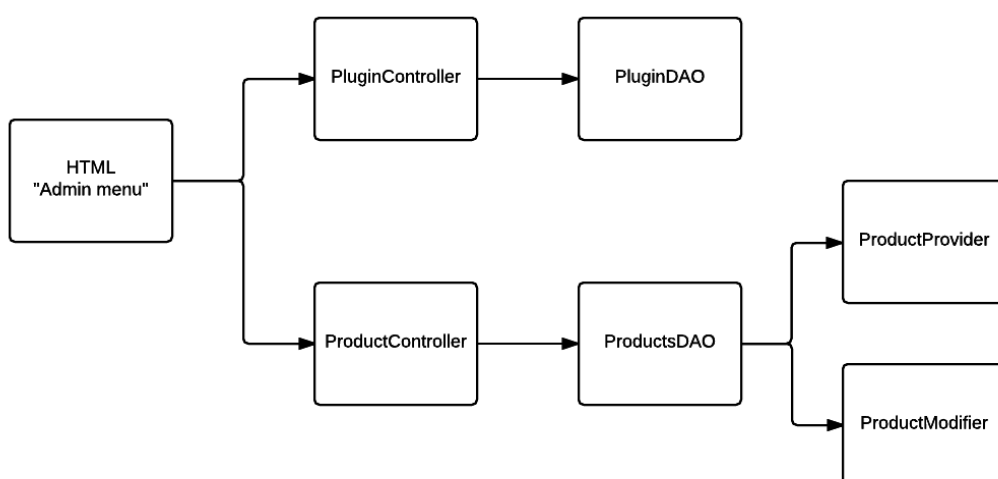


Figure 1 - Diagramme du patron DAO



Le diagramme ci-dessus démontre comment les classes du plug-in interagissent entre eux. Tout d'abord, lorsqu'on effectue une action par exemple l'affichage d'un produit, une requête provenant de la vue se fait traiter par le contrôleur responsable de gérer cette requête. Dans ce contrôleur, une série d'opérations permettront d'effectuer des appels à la base de données. Ensuite, ces opérations feront des appels à la base de données par le biais du *Data Access Object* dans lequel ce dernier contiendrait les objets qui feront ces appels.

Dans cette architecture, ces objets sont représentés par la classe « ProductProvider » qui contient les requêtes permettant de récupérer les produits et la classe « ProductModifier » qui contient les requêtes permettant d'écrire dans la base de données.

Pour ce qui est du « PluginDAO », celui-ci servait, lors de l'installation du plug-in, à créer les tables nécessaires pour conserver les configurations du plugin, ainsi que les tables qui contiendraient les informations des valves. Cependant, nous avons vite constaté que WordPress offrait la possibilité d'utiliser les tables wp\_post et wp\_postmeta (qui sont inclus dans l'installation de base de WordPress) pour conserver toutes les informations nécessaires au plugin. Ainsi, les classes PluginController et PluginDAO se sont avérés inutiles et n'ont pas été implémentés.

## Conception de l'interface utilisateur

### Mise en situation

Pour la partie de la conception de l'interface utilisateur du plug-in, Christian a créé les prototypes des différentes vues du plug-in. Lors de la création de ces interfaces, Christian a opté pour la création d'interfaces simples afin d'obtenir une meilleure utilisabilité pour le plug-in au client. Dans les sections ci-dessous, les étapes menant à la réalisation de l'interface utilisateur sera expliqué plus en profondeur.

### Besoins

Afin de répondre aux besoins du client, l'interface du plug-in devait être des plus simple d'utilisation possible. Donc, l'attribut de qualité le plus important, mis à part la

maintenabilité, était la convivialité. Ainsi, en tenant compte du niveau d'expérience du client, l'interface a été conçue de manière à minimiser la courbe d'apprentissage pour permettre au client d'atteindre une certaine autonomie le plus rapidement possible.

Chacune des fonctionnalités : ajout, modification et suppression d'un produit seront sur des pages séparées pour éviter toute confusion de la part de l'utilisateur. De plus, l'interface du plug-in devait être « responsive » pour qu'il puisse bien apparaître sous différentes résolutions d'écrans. Le « responsiveness » du plug-in a été possible grâce à la librairie Bootstrap CSS. Également, le plug-in devait respecter l'identité de la marque AJQ Services pour qu'il y ait une certaine cohésion visuelle. C'est pour cela que le logo d'AJQ Services se retrouve en permanence sur les pages du plug-in et la police de caractère et les couleurs choisies s'agencent bien avec ceux sur le site visible au public.

Finalement, concernant la langue du plug-in, seul l'anglais est supporté, car pour le moment, il n'y a qu'un seul utilisateur et il est anglophone donc, ce n'était pas nécessaire d'ajouter l'option de changer de langue.

## Prototype

Avant d'entamer l'implémentation du plug-in, nous avons préparé des prototypes des vues qui se retrouveront dans le plug-in. Il a fallu plusieurs itérations, et quelques rencontres avec le client pour arriver aux vues qui sont dans le plug-in actuellement. Par contre, les prototypes de départ n'étaient pas très loin de ce à quoi le client s'attendait. Chaque itération était une amélioration de ce qui avait été fait auparavant et non un changement complet de l'interface. Le plug-in comporte principalement deux vues : la vue principale qui affiche tous les produits ajoutés, la vue pour ajouter un nouveau produit.

Nous avons choisi de minimiser le nombre de vues non seulement pour faciliter la maintenance, mais aussi pour que ce soit simple aux yeux de l'utilisateur, car il aura

beaucoup moins d'interfaces dans lesquels naviguer et aussi, les fonctionnalités seront regroupées de manière cohérente.

### Ajouter un produit

L'interface pour l'ajout d'un produit a été simplifiée au maximum. L'utilisateur n'a qu'à saisir les informations au sujet du produit et puis en cliquant sur « save », le produit est sauvegardé dans la base de données de WordPress.

Tout d'abord, grâce au champ « product status » l'utilisateur peut indiquer s'il veut publier le produit immédiatement, ou bien il peut choisir de le créer en mode « draft » (brouillon). Lorsqu'un produit est créé en mode « draft », il apparaît dans la console administrateur, mais pas sur le site public. De plus, les champs « upload image » et « upload PDF » se servent du sélecteur média de WordPress. De cette manière, l'utilisateur n'a pas besoin d'apprendre plus d'une manière de téléverser ses fichiers et aussi, il n'a pas besoin de se soucier de l'endroit où déposer ses fichiers, car, tous les fichiers (que ce soit des images ou des PDF) qu'il aura téléversés seront dans le fichier Média de WordPress.

Également, une validation est effectuée lorsque l'utilisateur tente d'ajouter un nouveau produit. Le plug-in s'assure que le champ « product name » n'est pas vide avant de sauvegarder le produit. Finalement, un message apparaît pour signifier à l'utilisateur que l'opération s'est bel et bien déroulée.



## Valve Management

The screenshot shows a web interface for adding a product. On the left, there is a sidebar with a link to 'My Products' and a button labeled 'Add Product'. The main form area contains the following fields:

- Product name:** A text input field with the placeholder 'Name' and a small icon on the right.
- Product category:** A dropdown menu currently showing 'Automax'.
- Product status:** A dropdown menu currently showing 'Draft'.
- Product description:** A large, empty text area.
- Upload Image:** A text input field with the placeholder 'Image path...' and a blue 'Browse...' button.
- Upload Pdf:** A text input field with the placeholder 'Pdf path...' and a blue 'Browse...' button.
- Save:** A button located at the bottom right of the form.

*Figure 2 — Vue d'ajout de produit*

### Supprimer un produit

Pour ce qui est de l'interface pour la suppression d'un produit, l'utilisateur peut y accéder en allant tout d'abord sur l'onglet « My Products ». À partir de là, chacun des produits a un bouton « Delete ». En appuyant sur le bouton, une fenêtre modale apparaît pour que l'utilisateur confirme son action avant que le produit soit définitivement retiré de la base de données. Il est important de noter que tous les produits sont affichés dans l'onglet My Products, incluant aussi ceux qui ont le statut « draft ».

Finalement, dans le but de simplifier au maximum l'utilisation du plug-in, l'utilisateur a la possibilité de faire une recherche et/ou d'utiliser la pagination au bas de la page pour retrouver un produit plus rapidement afin de le supprimer ou bien de le modifier. Les produits peuvent aussi être filtrés par catégorie à l'aide du drop-down menu situé à gauche du champ de recherche.

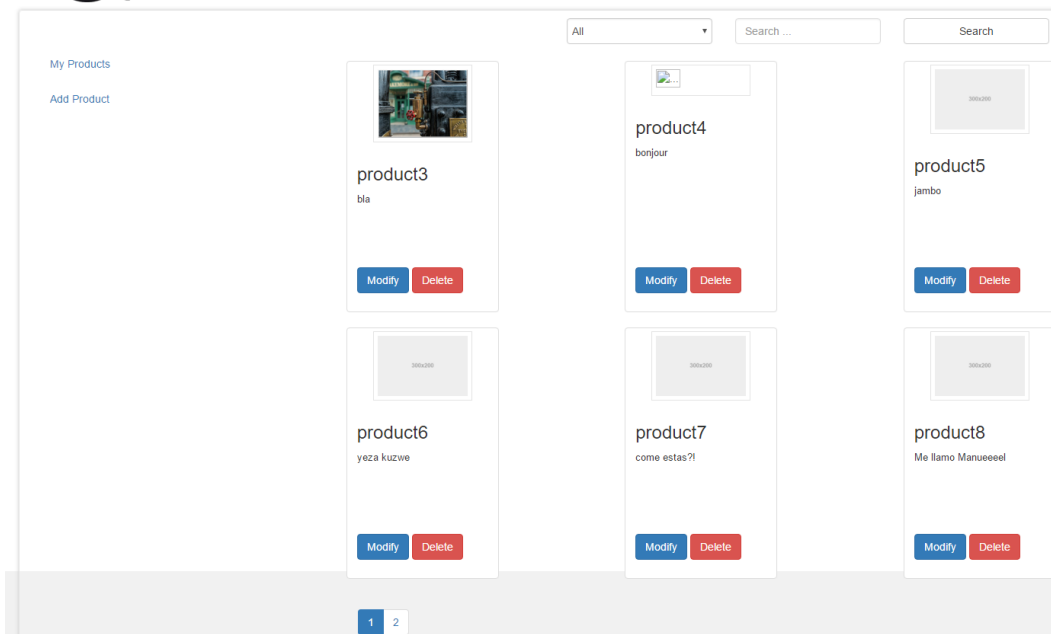
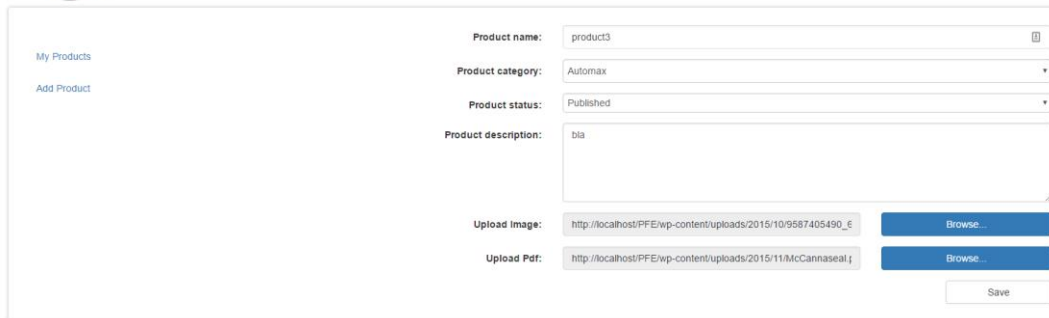


Figure 3 — Vue de tous les produits ajoutés

### Modifier un produit

En ce qui concerne l'interface pour la suppression de produit, il faut se rendre sur la page My Products. Ensuite, sur cette page, chaque produit a un bouton Modify (situé à côté du bouton Delete). Lorsque l'utilisateur appuie sur ce bouton, il est redirigé vers une page similaire à celle pour l'ajout d'un nouveau produit. Cependant, les champs sont déjà remplis avec les informations du produit sur lequel l'utilisateur avait cliqué. Ainsi, l'utilisateur peut modifier à sa guise chacune des informations concernant un produit. Une fois la modification complétée, il ne reste plus qu'à cliquer sur le bouton Save pour appliquer les changements à la base de données.

De plus, tout comme pour l'interface d'ajout d'un produit, une validation est effectuée lorsque l'utilisateur modifie un produit. Le plug-in s'assure que le champ « product name » n'est pas vide avant de sauvegarder les nouvelles informations d'un produit. Finalement, un message apparaît pour signifier à l'utilisateur que l'opération s'est bien déroulée.



My Products  
Add Product

Product name: product3

Product category: Automax

Product status: Published

Product description: bla

Upload Image:

Upload Pdf:

Figure 4 — Vue de modification d'un produit

## Discussion

La planification initiale des différentes vues manquait certaines fonctionnalités qui se sont avérées être essentielles au plug-in. Par exemple, la fonction de recherche, la pagination, la validation de formulaire et les messages de succès ou d'échec n'avaient pas été pris en compte lors de la conception initiale des interfaces. De plus, il y avait un onglet « Add Category » qui, finalement, ne servait à rien, car la fonctionnalité existait déjà dans WordPress. Également, les prototypes comportaient des onglets horizontaux pour filtrer les produits par catégories, mais cela a été remplacé par un drop-down menu. Enfin, malgré quelques différences, l'interface qui avait été prévue au départ a très peu changé par rapport à celle qui a été implémentée.

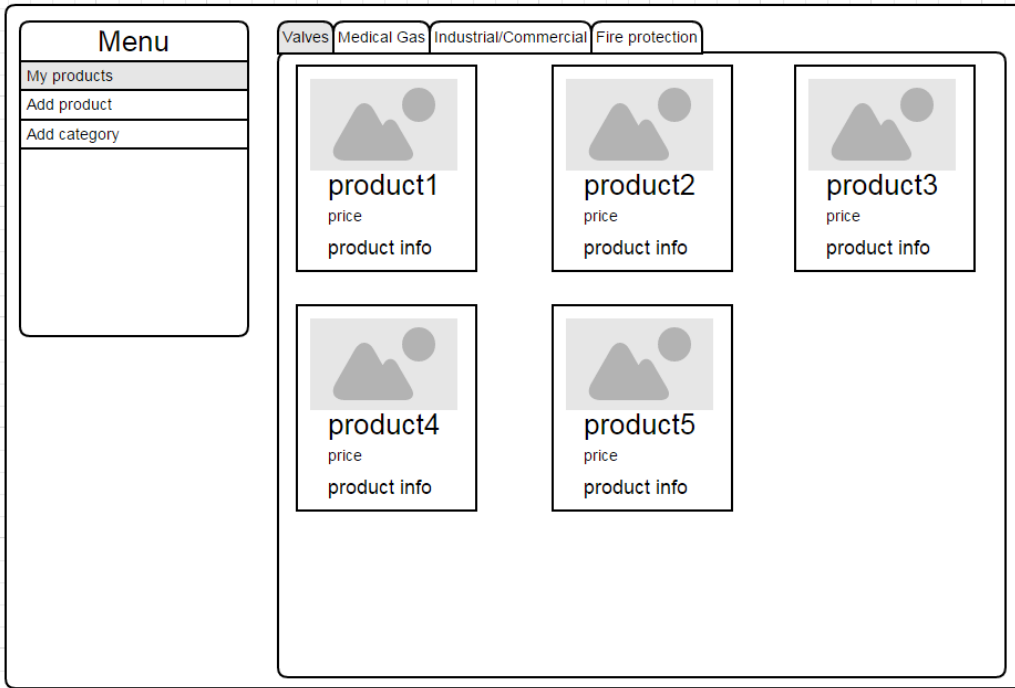


Figure 5 — Prototype initial

# Implémentation du plug-in

## Mise en situation

L'implémentation du plug-in s'est faite en deux étapes. Tout d'abord, l'implémentation de l'interface utilisateur a été faite par Gabriel et les bibliothèques utilisées sont principalement JQuery pour le traitement des requêtes AJAX et des différents événements déclenchés par le plug-in et Bootstrap pour les différents éléments visuels et styles présents dans le plug-in (boutons, onglets, etc.). Pour ce qui est de l'implémentation du plug-in, Christian s'est occupé de l'implémentation de l'affichage, l'ajout et la suppression d'un produit. Gabriel s'est occupé de l'implémentation de la modification d'un produit. Dans les sections ci-dessous, les différents logiciels, gabarits et bibliothèques utilisés y sont abordés.

## Logiciels utilisés

### Contrôle de version

Il existe plusieurs outils permettant de faire du contrôle de version. Pour les départager, il a fallu, en premier lieu, considérer ceux qui sont open source. Les plus populaires dans cette catégorie sont, en ce moment, SVN et Git. Le choix s'est arrêté sur Git principalement parce qu'il offre du contrôle de version décentralisé contrairement à SVN. En d'autres mots, Git permet à chaque développeur de travailler sur son poste et de faire des « commit » locales. De plus, il y a aussi le site GitHub qui permet de créer gratuitement un dépôt Git en ligne.

### Git

Le système de contrôle de version, Git, a été utilisé pour simplifier le développement. En effet, cet outil permet à plusieurs développeurs de contribuer à un projet en même temps sans se nuire et en plus, c'est facile de « merge » les différentes fonctionnalités. De plus, Git offre l'option de retourner à une version antérieure d'un projet, en gardant un historique de tous les changements qui ont été effectués dans le projet. Cela offre aussi un bon niveau de traçabilité, car tous les changements sont documentés.



## Environnement de développement

Pour ce qui est de l'environnement de développement, plusieurs choix étaient possibles. Nous aurions pu configurer un serveur Apache, puis installer MySQL et créer un schéma de base de données MySQL pour finir par installer PHP. Cependant, l'option la plus pratique était d'utiliser les utilitaires Wamp (Windows) et Mamp (Mac).

### Wamp/Mamp

Pour l'environnement de développement local, le choix s'est arrêté sur Wamp/Mamp pour la simplicité d'installation et d'utilisation. En effet, cet outil contient déjà tout le nécessaire pour mener à bien un projet de développement web en PHP et mysql. Ça évite le tracas d'avoir à configurer un serveur Apache ainsi qu'une base de donnée. En plus, cet utilitaire est open source, ce qui s'accorde très bien avec la licence « Creative Commons » du plug-in.

## Gestionnaire de contenu

En ce qui concerne le gestionnaire de contenu (CMS), nous étions obligés de respecter le choix que l'étudiant précédent avait fait. Son choix était judicieux contenu du fait que l'utilisateur n'est pas un expert et qu'une des préoccupations majeures était qu'il puisse être autonome le plus rapidement possible.

### WordPress

Plusieurs fonctionnalités de WordPress ont été utilisées pour le développement du plug-in. En effet, WordPress permet d'écrire des plug-ins tout en simplifiant plusieurs aspects comme la gestion et le chargement des scripts, la gestion des vues et des thèmes, ainsi que l'accès à la base de données. De plus, grâce à WordPress, ça n'a pas été nécessaire de créer tout un schéma de base donnée, car on peut utiliser les tables wp\_post et wp\_postmeta pour effectuer toutes les opérations du plug-in.

## Gabarits utilisés

### Gabarit visuel

Nous avons choisi de garder le gabarit que l'étudiant précédent avait choisi. Selon nous, « **Zerif-lite** » répondait bien aux besoins du site et visuellement, c'est un thème qui ajoute un air professionnel à un site informatif.

### Gabarit du plug-in

Nous nous sommes servis d'un gabarit pour développer le plug-in. Notre choix s'est arrêté sur « **WordPress-Plugin-Boilerplate** », car c'est un gabarit qui respectait déjà les normes et les conventions de développement tel que proposé par WordPress. Ainsi, nous n'avions qu'à modifier les différents fichiers du gabarit. En plus, la structure du gabarit était déjà bien faite alors nous avons pu sauver du temps à ce niveau-là.

## Librairies utilisées

Nous avons eu recours à quelques librairies pour nous faciliter la tâche pendant le développement du plug-in. Nous avons basé nos critères de sélections, pour les librairies utilisées, sur la disponibilité de la documentation ainsi que la facilité d'intégration avec WordPress. De plus, étant donné que le plug-in a été développé sous une licence Creative Commons, les librairies choisies devaient respecter les contraintes de cette licence.

### jQuery

jQuery nous a particulièrement été utile pour les requêtes Ajax. En effet, afin d'offrir au client une interface conviviale et facile d'utilisation, la plupart des pages du plug-in étaient chargées avec Ajax.

## Bootstrap CSS

Bootstrap CSS est une librairie CSS qui permet de créer des sites qui sont responsive. En d'autres mots, il permet de créer des sites dont l'apparence s'adapte à la résolution de l'écran sur laquelle le site est visité.

## SweetAlert

SweetAlert est une librairie JavaScript qui améliore l'apparence des « alert box » traditionnels. SweetAlert a aussi l'avantage de se centrer automatiquement sur la page, et d'être responsive.

# Formation sur la gestion du site web

## Mise en situation

La formation sur la gestion du site web se faisait par courriel ou bien lors des rencontres avec le client. Gabriel faisait la formation par courriel lorsqu'il y avait des problèmes puisqu'il avait un contact direct avec le client. Lors des rencontres, Gabriel et Christian s'occupaient de la formation de nouveaux éléments sur le site web et des fonctionnalités du plug-in. Quant à la réalisation du tutoriel sur le plug-in, c'est Christian qui s'est occupé de cette tâche. Les sections ci-dessous expliqueront les procédures utilisées lors de cette formation.

## Formation sur l'ensemble des fonctionnalités du site

### Assistance au client

Tout au long du projet, lorsque le client avait de la difficulté à opérer sur le site web, il pouvait nous contacter par courriel afin de pouvoir le dépanner en cas de problème. Parmi les problèmes qu'il a rencontrés, il y avait une situation où il ne pouvait pas effectuer de modifications sur le site web. Au début, on avait émis comme hypothèse qu'il aurait oublié comment se rendre à l'étape de personnalisation, mais ce n'était pas le cas. Après plusieurs essais, on s'est rendu compte que le client n'avait pas les droits nécessaires pour effectuer les

modifications. Donc en lui changeant ses accès, il a pu effectuer de nouveau ses modifications.

Lors de nos rencontres, on assistait aussi le client dans la gestion du site web. Donc si jamais il oubliait comment effectuer une action, il n'avait qu'à nous contacter par courriel ou par une rencontre afin de pouvoir résoudre ses problèmes.

## Tutoriel

Présentement, le tutoriel disponible concernant la gestion du site web concerne la création et l'utilisation de nouvelles sur le site web. Par contre, pour ce qui est de la personnalisation des éléments sur le site web ou pour l'ajout de nouveaux éléments sur le site web, il n'y a pas de tutoriel disponible. Donc, un tutoriel sera disponible afin qu'il puisse l'utiliser comme référence lorsqu'il aura des modifications à effectuer sur le site web.

## Formation sur le plug-in

### Tutoriel

Un tutoriel sur le fonctionnement des différentes fonctionnalités du plug-in sera disponible. Dans ce tutoriel, on retrouvera les sections concernant l'ajout, la modification, la suppression et la recherche de produits et ce dernier sera sous format PDF.

# Ajustement du contenu du site web

## Mise en situation

L'ajustement du contenu sur le site web a été fait par Gabriel. Les sections ci-dessous expliqueront les différents contenus qui ont été ajoutés ou modifiés sur le site web.

## Ajout du contenu

### Ajout des fichiers PDF

Afin que notre client puisse bien représenter les produits dont il fait la promotion, il a besoin d'un exemplaire .PDF de ces produits afin qu'il puisse les utiliser comme élément informatif pour les produits qu'il promeut sur le site web. Donc, on a reçu la demande d'ajouter les premiers fichiers .PDF afin qu'ils puissent être par la suite insérés dans le plug-in. Une fois les fichiers .PDF insérés, une formation a été donnée au client afin qu'il soit capable d'insérer des fichiers .PDF de façon autonome.

.

### Ajout des témoignages

Lors de la conception du site web, un plug-in sur l'affichage des témoignages était déjà présent sur le site en question. Par contre, lors d'une rencontre avec le client, il nous a fait remarquer que présentement, le plug-in permet d'afficher des témoignages, mais sous la forme de cartes. La problématique est qu'avec plusieurs témoignages, les nombres de cartes augmentent et cela ajoute de plus en plus de contenu sur la page web. Donc pour pallier à ce problème, un nouveau plug-in effectuant les mêmes tâches était nécessaire.

Pour ce faire, on a cherché dans le magasin de Wordpress un plug-in permettant de répondre aux besoins du client. Après la recherche, on a retenu le plug-in nommé « Testimonial rotator ». Ce plug-in permet d'afficher les témoignages sous la forme

d'un carrousel. Donc, en intégrant ce dernier dans le site web, nous sommes parvenus à répondre au besoin du client.

## Modification du contenu existant

### Modification des images du carrousel

Lors de notre première rencontre avec le client, ce dernier nous a mentionné qu'il voudrait que les produits dont il fait la promotion soient affichés dans le carrousel principal du site web. Afin de répondre à cette demande, on a dû utiliser l'éditeur graphique GIMP. Grâce à ce logiciel, on a pu prendre les différentes images de valves disponibles sur le site web, effectuer un montage, et créer de nouvelles diapositives avec les valves dont le client fait la promotion.

# Déploiement

## Mise en situation

Le déploiement a été fait par Christian grâce au logiciel Filezilla qui lui a permis de transférer les différents fichiers du plug-in dans l'environnement de « staging » et de « production ». La section ci-dessous expliquera les procédures pour y arriver

## Procédures

Pour ce qui en est du déploiement du plug-in, nous avons tout d'abord effectué des tests de fonctionnalités dans nos environnements locaux. Lorsque nous nous sommes assurés de la qualité du plug-in, nous avons mis en place un environnement de « staging ». Cet environnement est identique à l'environnement de production par contre, il ne s'agit pas de la même base de données. Nous avons créé une nouvelle base de données de « staging » puis, nous l'avons peuplée avec les mêmes données que ceux dans la base de données du site en production. Par la suite, nous avons activé le plug-in dans l'environnement de « staging ». Cette étape était nécessaire pour éviter de déployer le plug-in directement dans l'environnement de production et risquer de corrompre les données ou même que le site soit indisponible.

Une fois le plug-in installé dans l'environnement « staging », nous avons effectué d'autres tests fonctionnels pour être sûrs que tout fonctionnait encore comme prévu. Nous avons détecté quelques défauts que nous avons pu corriger immédiatement. Finalement, lorsque le plug-in dans l'environnement de « staging » était fonctionnel, nous l'avons installé dans l'environnement de production. Pour ce faire, nous avons utilisé l'outil FileZilla qui permet de téléverser des fichiers sur un serveur externe par le biais du protocole FTP.

## Conclusion

Dans le cadre de notre projet de fin d'études, nous avons conçu des prototypes et une architecture pour plug-in et ensuite, nous avons implémenté ce plug-in WordPress. De plus, nous avons interagi régulièrement avec un client réel tout au long du processus pour nous assurer de lui fournir un outil adéquat.

Ce projet a été très enrichissant tant au niveau académique que personnel. Tout d'abord, nous avons eu la chance de travailler avec un client qui n'était pas du domaine informatique. Nous avons dû adapter nos discours et aussi nos façons de faire afin qu'un client novice puisse se sentir en confiance quant à la gestion de son site. Ensuite, nous en avons appris beaucoup par rapport à la gestion d'un projet. Lors de la phase d'analyse de projet, nous avons établi un plan de travail avec une planification du temps requis pour accomplir toutes les tâches relatives à ce projet.

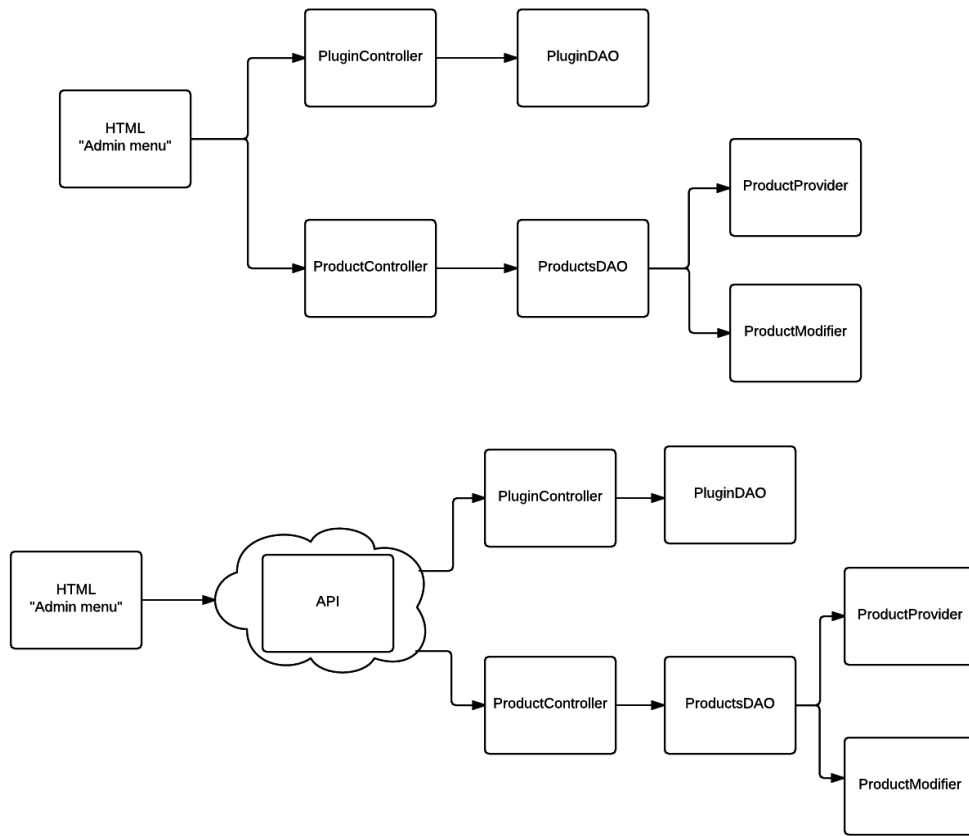
Cependant, nous avons très vite réalisé que les temps que nous avons estimés n'étaient pas réalistes. Nous avons sous-estimé certaines tâches qui nous ont ralenti dans l'exécution du projet. Un autre aspect important de ce projet de fin d'études a été la rédaction de document d'analyse de conception. En effet, nous avons rédigé un document de vision et un SRS pour bien cerner les besoins du client afin de lui offrir une solution adéquate. Finalement nous avons pu mettre en pratique d'autres notions apprises tout au long du bac comme l'utilisation de patrons de conception, et la mise en place d'une architecture logicielle.



## Recommandations

Lors de la conception de l'architecture du plug-in, il a été décidé que la logique d'affaires soit liée au site web pour des raisons d'implémentation. En effet, le fait d'avoir lié la logique d'affaire au site web nous a permis d'utiliser plus facilement les différents « hooks » de Wordpress lors de l'implémentation des fonctionnalités du plug-in. Par contre, pour ce qui est de la portabilité, cette conception ne serait pas adéquate. Si jamais pour une raison quelconque, on viendra à changer la technologie qui est utilisée par le site web par exemple un changement d'environnement, le plug-in au complet ne marchera pas.

Afin que le plug-in puisse être fonctionnel sous d'autres technologies, il faudrait créer un API qui aurait une architecture semblable à celui actuellement, mais qui ne serait pas lié complètement à la technologie de Wordpress. Cet API contiendrait les opérations nécessaires à l'exploitation de ce dernier et serait disponible à distance. Donc peu importe quelle technologie qui sera, utilisé pour l'implémentation de plug-in, l'API sera disponible afin d'offrir les opérations qui veillera au bon fonctionnement de ce dernier.



**Figure 5 — Diagrammes représentant l'architecture actuelle et faisant partie d'un API**

# Bibliographie

1. "Design Pattern Data Access Object Pattern." *www.tutorialspoint.com*. Web. 19 Apr. 2016.
2. Leffingwell, Dean, and Don Widrig. *Managing Software Requirements: A Use Case Approach*. Boston: Addison-Wesley, 2003. Print.
3. "WordPress.org." *WordPress › Support » Testimonial Rotator*. Web. 18 Apr. 2016.
4. "WordPress.org." *Writing a Plugin « WordPress Codex*. Web. 18 Apr. 2016.
5. Gabriel Pateau, Christian Bamatembera « Document de proposition » (Montréal), 01 Mars 2016
6. ETS, LOG410. « Analyse de besoins et spécifications » (Montréal), 22 janvier 2016.
7. Ross, Yvan. « Modele du domaine » (Montréal), 22 Janvier 2016.
8. Dupré-Corriveau, Maxime. « Solution Web pour AJQ Services INC. » (Montréal), 17 décembre 2015.