



Rapport de stage

Stéphanie Fournier

Février 2017

Cours 582-655-MA
Groupe 14653

Table des matières

DÉROULEMENT DU STAGE	3
1 L'organisme d'accueil du stagiaire	3
1a L'environnement de travail et les équipements.....	3
1b Le rôle et la fonction du stagiaire dans l'organisation	3
1c Le niveau de responsabilités	4
1d Les relations avec le milieu.....	4
2 Les activités du stage.....	4
2a Les travaux effectués (ampleur et complexité)	4
2b Les résultats obtenus.....	5
2c Le rendement fourni.....	5
3 Réflexion sur le déroulement du stage	6
4 Les acquis professionnels et humains.....	6
APPRENTISSAGES TECHNIQUES.....	7
1 Sujet technique	7
2 Problème technique	9
CONCLUSION	10
ANNEXES.....	12
Annexe A	13
Annexe B	14

DÉROULEMENT DU STAGE

1 L'organisme d'accueil du stagiaire

1a L'environnement de travail et les équipements

J'ai réalisé mon stage chez Openmind Technologies à Blainville. L'entreprise qui compte un peu plus de 25 employés offre des services informatiques aux entreprises ainsi que des services de développement Web. D'après mes observations, les trois principaux pôles Web de l'entreprise sont le développement, le commerce électronique et la gestion d'hébergement.

Le travail des développeurs Web et programmeurs se passe dans un environnement PC avec Windows 10. Les principaux logiciels et outils que j'ai été amenée à utiliser sont :

- GapDebug
- Git / GitLab
- Google (Gmail, Agenda et Drive)
- HeidiSQL
- SourceTree
- Trello (gestion Agile)

Du point de vue technique, j'ai utilisé :

- Cadres d'application (*frameworks*)
Ionic et Laravel
- jQuery .validate
- MariaDB
- Metronics
(thème basé sur Bootstrap)
- WordPress

1b Le rôle et la fonction du stagiaire dans l'organisation

Dès ma première journée, j'ai été intégrée à l'équipe au même titre que les employés permanents, c'est-à-dire qu'on m'a assigné de vrais projets. J'ai donc eu un rôle de développeuse Web, lequel m'a permis de faire de l'intégration et du développement.

1c Le niveau de responsabilités

Malgré mon statut de stagiaire, mon niveau de responsabilité était comparable aux autres développeurs Web. La seule règle plus souple a été au niveau des délais pour réaliser les projets, surtout au début de mon stage. Dès le début lorsqu'on m'assignait des tâches, on m'indiquait le temps de travail qui avait été évalué pour chaque tâche, mais mes patrons étant conscients que j'étais encore en apprentissage, ils ne m'ont jamais mis de pression pour respecter de façon stricte ces délais.

1d Les relations avec le milieu

L'entreprise a mis sur pied plusieurs outils afin de faciliter l'interaction entre les employés. Un des outils que j'ai trouvé particulièrement pratique est la messagerie *Slack* puisque celle-ci permet autant les conversations privées que les canaux de discussion. Par exemple, nous avons un canal « Bonnes nouvelles » qui permet à chacun de partager à toute l'entreprise ses bonnes nouvelles professionnelles ou personnelles.

De plus, chaque semaine une courte réunion est prévue avec le président et tous les employés afin de faire le point sur les projets internes, propager les bonnes nouvelles, souligner les bons coups et transmettre toute autre information pertinente à l'équipe.

2 Les activités du stage

2a Les travaux effectués (ampleur et complexité)

Mon stage a été divisé en deux parties, la première moitié a été consacrée à l'intégration (front-end) alors que je me suis concentrée sur la programmation (back-end) dans la seconde moitié.

Les travaux effectués dans la portion intégration de mon stage étaient de complexité simple à moyenne, puisque c'était majoritairement de l'intégration dans des sites existants. Je peux donc dire que c'étaient des projets de plus petite envergure à l'exception du site Web Hidden Place que j'ai complètement refait pour le mettre en WordPress.

Dans la portion programmation de mon stage, j'ai travaillé sur un seul projet, mais de plus grande envergure, soit la plateforme client de Bye Bye Bail. J'ai réalisé diverses tâches dans ce projet allant de tâches simples comme la réalisation d'un tableau de bord à des tâches beaucoup plus complexes comme l'implantation d'une messagerie.

2b Les résultats obtenus

Toutes les tâches qui m'avaient été attribuées dans la portion intégration de mon stage ont été réalisées avec succès. Le site Web de Virginie Coossa a été lancé à la date prévue et la cliente en était très satisfaite. Le chargé de projet responsable de l'application et du site de Hidden Place a aussi été très satisfait du résultat de l'intégration faite sur ces deux plateformes.

Du côté programmation, une seule des tâches à faire n'a pas été complétée puisque des exigences supplémentaires se sont ajoutées ce qui a prolongé le temps de développement.

2c Le rendement fourni

Lors des deux ou trois premiers jours du stage, j'avais l'impression d'être beaucoup plus lente que les autres, mais après avoir apprivoisé les principaux logiciels utilisés, je considère avoir atteint un niveau d'efficacité d'environ 85 à 90 % par rapport aux intégrateurs en place.

Pour ce qui est de la portion programmation, comme je ne connaissais pas du tout Laravel, ça m'a pris un peu plus de temps avant d'avoir un rendement aussi élevé. Si j'évalue mon rendement à la fin de mon stage, soit après seulement douze jours

de programmation, je considère me situer à environ 80 ou 85 % du rendement des autres programmeurs.

Parmi les outils et technologies que j'ai utilisés et que je n'avais pas vus lors de la formation mentionnons :

- GIT
- jQuery .validate
- Laravel
- WordPress :
 - Utilisation de thèmes achetés
 - Extensions : Visual Composer, Contact Form 7, Revolution Slider, Loco Translate

3 Réflexion sur le déroulement du stage

Je me considère chanceuse d'avoir pu faire de l'intégration et de la programmation au cours de mon stage. Au début, j'avais un peu peur que six semaines ça ne soit pas assez long pour vraiment prendre le pouls du marché du travail, mais finalement c'est suffisant pour voir de façon réaliste le travail quotidien d'un développeur Web.

4 Les acquis professionnels et humains

Même si au cours de la formation nous sommes régulièrement appelés à travailler en équipe, la réalité sur le terrain est un peu différente. C'est-à-dire que, oui, nous travaillons en équipe, mais plutôt de façon parallèle et sur des tâches prédéterminées. J'ai donc appris en cours de stage à me faire plus confiance pour avancer mes tâches tout en allant chercher l'aide de mes collègues au besoin.

Puisque six semaines, c'est très vite passé, j'ai aussi pris conscience de l'importance de s'intégrer rapidement à une équipe. Ça facilite grandement le travail d'équipe et ça rend la vie au bureau beaucoup plus agréable.

APPRENTISSAGES TECHNIQUES

1 Sujet technique

Au cours de mon stage, j'ai commencé à utiliser le cadre d'application (*framework*) Laravel. Ce cadre est basé sur le principe modèle-vue-contrôleur (MVC), mais dans une version beaucoup plus complexe que les MVC avec lesquels j'avais travaillé auparavant.

Puisque le projet était déjà débuté depuis un bon moment et que plusieurs fonctionnalités étaient déjà en place, j'ai pu étudier ce qui avait été fait par les autres développeurs afin de bien comprendre le fonctionnement de ce nouvel outil.

Ce que j'ai trouvé particulier au départ a été le fait de devoir déclarer les routes dans des fichiers « routes ». Une fois que l'on s'habitue à inscrire ses routes dans ces fichiers, il est très simple de les retrouver puisqu'elles sont toutes au même endroit. C'est, selon moi, un grand avantage de ce cadre d'application.

Une autre particularité non négligeable de Laravel est l'utilisation du système « Eloquent ORM » (Object Relational Mapper). L'avantage de ce système est de faire des requêtes qui s'adapteront ensuite à tous les systèmes de gestion de base de données.

Par exemple, si dans l'environnement de développement vous utilisez une base de données MariaDB, mais que lorsque vous mettez le site en production chez le client celui-ci utilise MySQL, toutes les requêtes écrites avec l'ORM continueront de fonctionner sans intervention manuelle du programmeur. Ça permet donc d'économiser du temps et par le fait même des coûts de développement.

Afin d'utiliser le système « Eloquent ORM », il faut d'abord créer un modèle de base qui sera ensuite utilisé dans les modèles « enfants » contenant les requêtes.

Le modèle Eloquent utilise certaines constantes qui seront automatiquement ajoutées aux requêtes. Parmi celles-ci, mentionnons « `CREATED_AT` » et « `UPDATED_AT` » qui inséreront une date sous forme de « timestamp » dans les requêtes d'insertion ou de mises à jour (insert et update). Il est possible de définir le nom des colonnes où ces informations seront ajoutées soit dans le modèle « parent » Eloquent ou encore de les redéfinir dans les modèles « enfants ».

La syntaxe des requêtes écrire avec l'ORM de Laravel est plutôt particulière, puisqu'elle se fait en style orienté objet. Le premier paramètre est la table principale de la requête, ce paramètre est défini par la propriété « `table` » au début de chaque modèle « enfant ». On utilise donc habituellement le *\$this* pour faire référence à cette table. La requête se poursuit avec les jointures, les conditions (*where*), la sélection des colonnes (*select*) et finalement l'exécution (*get* pour toutes les lignes ou *first* pour le premier résultat seulement).

([Annexe A](#))

Pour les requêtes plus complexes, Laravel permet d'utiliser des expressions brutes (*Raw Expressions*) avec la syntaxe « `DB::raw ()` ». Cette façon de faire est particulièrement pratique pour faire une requête sur une table différente de celle définie dans le modèle. On utilise alors l'expression : « `DB::table(table)` ».

Les expressions brutes sont aussi utiles lors de requêtes complexes, par exemple pour une requête qui contiendrait les conditions suivantes :

WHERE condition1 AND (condition2 OR condition3)

([Annexe B](#))

Malgré le fait que je ne puisse comparer Laravel aux autres cadres d'application existants, je trouve qu'il s'agit d'un excellent système qui permet d'avoir rapidement et facilement une bonne structure MVC. C'est donc une technologie que je recommanderais pour tout site Web le moins complexe.

2 Problème technique

Les problèmes techniques plus complexes que j'ai rencontrés en cours de stage étaient reliés à Laravel, au PHP et à jQuery/Ajax. J'ai entre autres perdu du temps à plusieurs reprises pour transformer des requêtes complexes écrites en SQL vers le format spécifique de l'ORM de Laravel.

Le plus grand problème que j'ai rencontré concerne les dates dans le projet réalisé en Laravel. Tout d'abord, je devais calculer l'écart entre le moment présent et une date enregistrée dans la BD, mais mon problème était que les 2 dates n'étaient pas dans le même fuseau horaire, donc au lieu de sortir un écart de 15 minutes j'obtenais un écart de 5 heures !

J'ai tenté de chercher un peu sur Internet, mais c'était plutôt flou alors j'ai discuté avec un collègue et c'est celui-ci qui m'a expliqué que le problème était notre BD qui n'était pas en temps « UTC ». Il m'a donc fourni les informations pour corriger le problème dans la BD de même que dans le fichier d'accès à la BD.

Cette solution a par contre causé un autre problème lors des derniers jours de mon stage. J'ai voulu afficher la date d'enregistrement d'un message dans une sous-vue appelée en AJAX, mais l'heure affichée est dans le fuseau horaire « UTC ». Malgré mes essais à utiliser la fonction « strftime() » qui devrait formater la date et l'heure selon la localisation locale, l'heure restait dans le fuseau horaire « UTC » et la date s'affichait encore en anglais malgré le fait que la localisation ait été réglée en utilisant la fonction « setlocale() ». Le problème semblait venir de l'appel de la sous-vue en AJAX.

Comme le problème est survenu à la fin de mon stage, je n'ai pas pu le résoudre, mais ça sera mon premier défi en tant qu'employée. J'irai chercher conseil auprès de mes collègues pour tenter de trouver une solution.

CONCLUSION

Ouf ! Cette réflexion sur le stage me fait aujourd'hui réaliser tout le chemin parcouru depuis le début de ma formation au Collège. Que d'efforts mis depuis un an et demi pour en arriver au résultat espéré : un travail dans le Web ! Je dois avouer que je suis extrêmement fière de mes accomplissements et de l'endroit où je suis rendue maintenant.

C'est vrai que la formation est très intense et demande beaucoup d'efforts et d'implication, mais je trouve que les efforts requis ne sont pas exagérés. Comme le dit l'expression « dans la vie rien n'est gratuit », le prix à payer ici étant de l'investissement de temps personnel qui à la fin est bien payant.

Je crois que la formation offre une préparation adéquate pour débiter dans le monde de la conception et du développement Web. Évidemment, en moins de deux ans on ne peut pas devenir des experts, mais cela se fera au fil de nos expériences de travail. Rien ne vaut la pratique !

Malgré cette préparation, j'ai trouvé qu'il y a un écart plutôt important entre la formation et la réalité sur certains points, particulièrement sur le design. Au cours de la formation, on nous laisse presque toujours le loisir de décider nous-mêmes du design de nos sites. Je crois qu'il faudrait plus souvent nous mettre des contraintes pour le visuel comme nous le vivons au travail. En effet, souvent un graphiste fait les maquettes et en tant qu'intégrateur nous nous devons de les reproduire au pixel près. Comme cela ne nous est presque jamais demandé dans la formation, c'est une adaptation majeure à faire à nos méthodes de travail une fois en emploi.

L'autre critique que j'aimerais faire par rapport à la formation est qu'on devrait plus souvent nous exiger de réaliser des travaux individuels. Je crois qu'il est trop facile pour certains de réussir leur cours en se fiant au travail de leur coéquipier lors des travaux d'équipe. Une piste de solution serait peut-être de demander des travaux d'équipe, mais que ce soit l'enseignant qui se charge de répartir les tâches (comme le patron qui répartit les tâches). Ainsi chaque coéquipier serait noté sur la portion du travail qui lui était attribuée. Les étudiants pourraient ainsi s'entraider tout de même, mais les notes seraient plus justes pour chacun.

J'ai réalisé lors de la rencontre post-stage la différence entre les étudiants ayant trouvé leur stage eux-mêmes et ceux ayant pris un stage offert par le Collège. Ayant moi-même fait les démarches pour dénicher mon stage, je crois que cette démarche est très importante et qu'elle devrait être encouragée par le Collège. Lorsqu'on nous a parlé des stages, on nous a mentionné cette possibilité, mais comme en même temps on nous a demandé de ne pas court-circuiter le processus mis en place par le Collège ça portait à confusion. Je considère qu'il serait bon de sensibiliser les étudiants à l'importance de faire leurs propres recherches. Ce processus devrait même être encadré par le Collège et du temps devrait être prévu à l'horaire pour nous accompagner dans ces démarches.

En conclusion, j'ai eu une très belle expérience de stage et malgré les quelques désagréments une belle expérience générale de formation au Collège Maisonneuve.

ANNEXES

Annexe A

Requête simple utilisant la syntaxe de *Eloquent ORM*

```
/**
 * Get leases cities informations
 *
 * @return array
 */
public function getLeasesByCities()
{
    $output = $this->join('clients', 'leases.client_id', '=', 'clients.id')
        ->where('leases.lease_status_id', '=', DB::raw('LEASE_STATUS_LIVE'))
        ->select(['clients.city', DB::raw('COUNT(leases.id) as nb')])
        ->groupBy(['clients.city'])
        ->orderBy('clients.city', 'ASC')
        ->get();

    return $output;
}
```

Annexe B

Requête complexe utilisant la syntaxe de *Eloquent ORM* et les expressions brutes (voir l'encadré)

```
/**
 * Get all information requests and date of last message for a user (messages page)
 */
* @param int $clientId
* @param string null $lang
* @return mixed
*/
public function getContactsList($clientId, $lang = null)
{
    if (empty($lang)) {
        $lang = App::getLocale();
    }

    $output = $this->leftJoin('leases', 'informations.lease_id', '=', 'leases.id')
        ->leftJoin('clients', 'clients.id', '=', 'informations.client_id')
        ->leftJoin('clients as seller', 'leases.client_id', '=', 'seller.id')
        ->leftJoin('vehicles', 'vehicles.id', '=', 'leases.vehicle_id')
        ->leftJoin('models_brands', 'vehicles.model_brand_id', '=', 'models_brands.id')
        ->leftJoin('models', 'models_brands.model_id', '=', 'models.id')
        ->leftJoin('brands', 'models_brands.brand_id', '=', 'brands.id')
        ->leftJoin(DB::raw('(SELECT MAX(messages.created) as messCreated, messages.information_id FROM
            $join->on('mess.information_id', '=', 'informations.id');
        ))
        ->select(['leases.client_id as leasesClientId',
            'informations.id as infoId',
            'informations.client_id as infoClientId',
            'informations.lease_id as adId',
            'informations.name as infoName',
            'informations.status_buyer as infoStatusBuyer',
            'informations.status_seller as infoStatusSeller',
            'informations.read_buyer as readBuyer',
            'informations.read_seller as readSeller',
            DB::raw('CONCAT (clients.firstname, " ", clients.lastname) as clientName'),
            DB::raw('CONCAT (seller.firstname, " ", seller.lastname) as sellerName'),
            'brands.name as brand',
            'models.name as model',
            'vehicles.version',
            'models_brands.year',
            'mess.information_id as messInfoId',
            'mess.messCreated as lastMess'])
        ->orderBy('mess.messCreated', 'DESC')
        ->where(function ($query) use ($clientId) {
            $query->where('informations.client_id', '=', $clientId)
                ->where('informations.status_buyer', '=', INFORMATIONS_STATUS_ACTIVE);
        })
        ->orWhere(function ($query) use ($clientId) {
            $query->where('leases.client_id', '=', $clientId)
                ->where('informations.status_seller', '=', INFORMATIONS_STATUS_ACTIVE);
        })
        ->get();

    if ($output->count() > 0) {
        return $output;
    } else {
        return false;
    }
}
```