



POLYTECHNIQUE
MONTREAL
LE GÉNIE
EN PREMIÈRE CLASSE



RAPPORT DE STAGE

Déploiement des services d'analyse de qualité logiciel vers les serveurs de Polytechnique Montréal et développement d'une interface Web utilisateur intuitive



Réalisé par

Arnaud SOULIER

Sous la direction de

Yann-Gaël Guéhéneuc

Superviseur de stage

Francis Garcia

Tuteur de stage

Pour l'obtention du DUT Informatique

Année universitaire 2015 – 2016

Remerciements

Je tiens à remercier toutes les personnes qui ont rendu possible cette expérience et qui m'ont permis de faire mon stage dans les meilleures conditions.

Tout d'abord j'adresse mes principaux remerciements à mon maître de stage ainsi qu'à mon tuteur de stage. Le professeur Yann-Gaël Guéhéneuc m'a accepté dans son équipe de recherche et permit de vivre cette fabuleuse expérience. Le professeur Francis Garcia m'a accompagné dans mes démarches. Parallèlement, je remercie aussi le professeur Foutse Khomh qui m'a accueilli à mon arrivée sur place et soutenu durant mon stage.

Je tiens à remercier vivement Mme Johanne Lemieux, pour m'avoir aidé avec beaucoup de gentillesse dans les démarches administratives pour l'obtention du permis de travail et toutes autres démarches et documents à remplir.

Je remercie également toutes les personnes travaillant au SoccerLab pour m'avoir accueilli et avoir facilité mon intégration ainsi que pour leur soutien et l'entraide dont ils ont fait preuve pendant ces 3 mois.

Enfin je tiens à apporter mes remerciements à tous mes proches qui m'ont encouragé et soutenu dans cette aventure. J'ai une pensée particulière envers Sébastien dont les conseils m'ont beaucoup aidé dans la recherche de mon stage, ainsi qu'à mon camarade Gabriel, qui m'a gentiment offert ces services de designer.

Sommaire

Introduction	1
1 Présentation de l'entreprise.....	2
1.1 Polytechnique Montréal.....	2
1.2 Département Génie Informatique et Génie Logiciel	2
1.2.1 PolyMORSE	3
1.2.2 Ptidej Team	3
2 Cahier des charges	4
2.1 Mission	4
2.2 Besoins fonctionnels.....	4
2.3 Besoins non-fonctionnels	4
2.3.1 Outils et langages	4
2.3.2 Autres contraintes	5
3 Rapport technique.....	6
3.1 Conception.....	6
3.1.1 Application Web	6
3.1.2 Mise à jour des fonctionnalités	6
3.1.3 Base de données	6
3.2 Résultats et perspectives	7
3.2.1 Interfaces utilisateur	7
3.2.2 Perspective	7
4 Manuel d'utilisation	9
4.1 Gestion des utilisateurs	9
4.1.1 Création d'un compte utilisateur	9
4.1.2 Connexion	10
4.1.3 Modification de vos paramètres et suppression du compte	10
4.2 Gestion des projets.....	11
4.2.1 Création d'un projet.....	11

4.2.2 Suppression d'un projet.....	12
4.3 Gestion des versions	12
4.3.1 Ajout d'une version	12
4.3.2 Suppression d'une version.....	13
4.4 Appel d'un service.....	14
4.5 Gestion administrateur.....	15
5 Rapport d'activité	16
5.1 Découverte.....	16
5.2 Faire marcher le travail d'avant	16
5.3 Site Web	17
5.4 Base de données.....	17
5.5 Finition.....	17
Conclusion.....	19

Table des figures

Figure 1 - Polytechnique Montréal.....	2
Figure 2 - Emblème Polytechnique	2
Figure 3 - Bannière Ptidej.....	3
Figure 4 - Formulaire création utilisateur.....	9
Figure 5 - Affichage création utilisateur	9
Figure 6 - Formulaire connexion.....	10
Figure 7 - Formulaire mise à jour mot de passe	10
Figure 8 - Affichage mise à jour mot de passe	10
Figure 9 - Formulaire suppression utilisateur.....	11
Figure 10 - Affichage suppression utilisateur	11
Figure 11 - Formulaire création projet	11
Figure 12 - Affichage création projet	11
Figure 13 - Formulaire suppression projet	12
Figure 14 - Affichage suppression projet	12
Figure 15 - Formulaire création version.....	12
Figure 16 - Affichage création version.....	13
Figure 17 - Formulaire suppression version.....	13
Figure 18 - Affichage suppression version.....	13
Figure 19 - Formulaire appel services	14
Figure 20 - Affichage appel service POM.....	14

Glossaire

Mitigation des risques par "tampon" : méthode de gestion des risques où l'on planifie la durée d'une tâche donnée, puis, pour pallier à tout problème, on multiplie cette durée par deux ou par trois.

Java EE : Java Enterprise Edition, version de Java destiné au développement d'application dynamique en ligne.

OpenStack : outil permettant d'émuler des instances de machine virtuelle et les rendre accessible comme sur un Cloud.

Méta-modèle : concept présentant la structure d'un modèle en programmation orienté objet.

Gradle : moteur de production permettant de construire des projets avec de nombreuses dépendances.

Introduction

Depuis de nombreuses années, le secteur de l'informatique a subi de fortes évolutions. De nouvelles technologies apparaissent et de plus en plus de codes et de programmeurs voient le jour. Chaque codeur a sa propre façon de coder.

Cependant, toutes les manières ne sont pas jugées propres et/ou optimisées. Il existe une grande quantité de conventions permettant aux programmeurs du monde entier d'être en accord sur différentes règles de codage. De la simple convention de nommage à la convention d'architecture de programme, elles ne sont pas obligatoires mais dans un souci de compréhension mutuelle, elles sont fortement conseillées. De plus, certaines conventions se prêtent mieux à certains programmes que d'autres, le choix peut donc avoir un impact sur l'évolution (future) d'un programme. Le concept de "dette technique" qui émerge est directement lié à celui de la qualité logicielle.

Le professeur Yann-Gaël Guéhéneuc a donc décidé de développer *Ptidej*, une suite d'outils permettant de calculer et d'évaluer la qualité d'un système logiciel à l'aide de métriques (mesures logicielles) et de " patrons " qui représentent des conventions de programmation, de conception et d'architecture. L'objectif de ce projet sera de rendre accessible à tous, les services d'analyse de qualité logiciel *Ptidej* par le biais de l'application Web *Brunch*.

Ce rapport se divise en cinq parties. Dans un premier temps, je présenterai l'entreprise d'accueil de mon stage. J'annoncerai alors les différentes contraintes fonctionnelles et non-fonctionnelles du cahier des charges. Viendra ensuite la présentation de la conception et du développement du projet avec le rapport technique avant d'expliquer son fonctionnement dans un manuel d'utilisation. Pour finir, je présenterai la planification de mon travail et son déroulement dans un rapport d'activité.

1 Présentation de l'entreprise

1.1 Polytechnique Montréal

Fondée en 1878, l'Université de Montréal, avec ses deux écoles affiliées, l'École Polytechnique et la HÉC Montréal, est aujourd'hui la plus grande université du Québec et la seconde plus grande du Canada. Elle est également la plus grande université francophone extérieure à la France. Avec plus de 60,000 étudiants du monde entier, plus de 11,000 diplômes attribués chaque année. Ces dépenses dans la recherche sont de plus d'un demi-milliard de dollars, ce qui en fait la troisième université canadienne la plus dynamique en termes de recherche. C'est également la plus dynamique du Québec depuis 2005.

Fondée en 1873 par Urgel-Eugène Archambault, l'École Polytechnique de Montréal est l'une des trois plus grandes écoles d'ingénierie du Canada et la plus grande du Québec. C'est l'une des plus grandes institutions d'enseignement et de recherche en ingénierie du pays. Son domaine d'enseignement couvre une grande partie des domaines scientifiques au sein de sept départements.



Figure 1 - Polytechnique Montréal



Figure 2 - Emblème Polytechnique

Son emblème représente une abeille au milieu d'un engrenage traversé par une poutre d'acier, le tout entouré d'une couronne de laurier. L'abeille représente le travail organisé et planifié des ingénieurs. La poutre en acier symbolise le premier département qui a vu le jour au sein de l'École Polytechnique de Montréal : le génie civil. L'engrenage suggère la montée industrielle à la fin du 19^e siècle dans laquelle les ingénieurs ont joué un rôle important. Enfin, la couronne de laurier symbolise l'excellence. Cet emblème est accompagné de la devise "Ut Tensio Sic Vis" qui se fonde sur une loi de résistance matérielle appelée "La loi de Hooke". Sa traduction littérale signifie "L'allongement est proportionnel à la force". Au sens figuré, cela signifie "les résultats sont proportionnels aux efforts apportés". C'est une devise qui inspire tous les polytechniciens.

1.2 Département Génie Informatique et Génie Logiciel

Le département Génie Informatique et Génie Logiciel (GIGL) regroupe de nombreuses équipes dans divers laboratoires de recherche. Chaque professeur du département dirige un laboratoire. Certains se sont regroupés autour d'un thème commun, le génie logiciel, et ont formé le groupe de recherche PolyMORSE.

1.2.1 PolyMORSE

Le groupe PolyMORSE (POLYtechnique MONtréal Researchers in Software Engineering) est un groupe de recherche du département GIGL regroupant cinq des plus gros laboratoires de recherche du département :

- **SOCCER** : Software Cost-effective Change and Evolution Research
- **SWAT** : SoftWare ANalytics and Technologies Lab
- **RGL** : Software Engineering Research
- **MCIS** : Maintenance, Construction and Intelligence of Software
- **Ptidej Team** : Pattern Trace Identification, Detection, and Enhancement in Java

1.2.2 Ptidej Team

L'équipe de *Ptidej* (Pattern Trace Identification, Detection, and Enhancement in Java) a pour but de mettre au point des théories, méthodes et outils pour comprendre, évaluer et améliorer la qualité des logiciels. Pour cela, ils promeuvent l'utilisation de patrons et d'architectures prédéfinis. Ils veulent formaliser les patrons pour identifier des occurrences de patrons, puis améliorer ces occurrences. Ils étudient et évaluent également l'impact de ces patrons sur la qualité de systèmes logiciels.

Divers outils ont été développés, notamment la suite d'outils *Ptidej*, afin d'évaluer et d'améliorer la qualité des systèmes logiciels en favorisant l'utilisation de patrons au niveau du langage, de la conception ou de l'architecture.



Figure 3 - Bannière Ptidej

2 Cahier des charges

Étant donné qu'il s'agit ici d'un projet de recherche, il n'y a pas de cahier des charges précis établi. Cependant, certaines contraintes et tâches sont tout de même à respecter.

2.1 Mission

Le projet consistera à déployer des services déjà existants. En effet, la suite d'outils *Ptidej* est fonctionnelle et l'application Web *Brunch* existe à l'état de services. Il faudra donc la rendre totalement opérationnelle, la déployer sur les serveurs de l'École et, pour cela, finir l'implantation des services et concevoir, implanter et déployer le code nécessaire pour rendre accessible ces services depuis l'Internet.

2.2 Besoins fonctionnels

Le projet comporte trois grandes lignes de réalisation. Ces trois tâches forment donc le fil rouge du projet.

Dans un premier temps, il faudra finaliser l'application Web précédemment développée. Cette tâche demandera une relecture du code déjà présent ainsi qu'une correction d'éventuelles erreurs de compilation ou de développement.

Ensuite, une fois les services de l'application en état de marche, il faudra la déployer sur les serveurs de l'École afin de la rendre accessible depuis l'extérieur. Il faudra que n'importe qui dans le monde puisse accéder aux services.

Pour finir, il faudra développer une interface utilisateur pour simplifier l'utilisation de l'application et la rendre accessible par tous. Cette interface prendra l'apparence d'un site Web.

Comme indiqué sur le diagramme des cas d'utilisation (cf. Annexe 1), l'utilisateur devra pouvoir créer, afficher et supprimer un utilisateur, un projet logiciel et une version depuis l'application. Il devra également pouvoir faire appel aux différents services disponibles.

2.3 Besoins non-fonctionnels

2.3.1 Outils et langages

Afin de procéder au déploiement, le serveur *GlassFish* devra être utilisé pour héberger l'application, développée en Java EE*. Ce serveur pourra être exécuté sur une instance d'*Openstack** ou bien directement sur une instance d'une machine virtuelle sur un serveur de l'École. Le choix des langages et des technologies utilisés pour le développement de l'interface utilisateur est libre.

2.3.2 Autres contraintes

L'application devra intégrer un système de sécurité afin de protéger l'accès aux services. N'importe qui a le droit d'accès mais le code doit tout de même être protégé.

Pour finir, le code devra être commenté et rédigé en anglais, tout comme la totalité du site Web.

3 Rapport technique

3.1 Conception

3.1.1 Application Web

Le projet *Ptidej* est une suite d'outils en Java permettant d'évaluer la qualité de programmes orientés objet. Il est constitué de plusieurs modules pour un total actuel de plus de 2,000 classes. C'est un projet très complexe. Dans son fonctionnement, il commence par créer un modèle du code qu'il reçoit, modèle conforme à un méta-modèle* similaire au méta-modèle UML, avant d'effectuer des analyses dessus. Il possède de nombreux services mais seuls certains sont pour l'instant extraits vers *Brunch*.

Brunch est une application Web. Elle utilise la technologie Java EE ainsi que l'API *REST* pour donner un accès aux services de *Ptidej* et le cadriciel *Jersey*. Tous deux permettent l'utilisation des méthodes du protocole HTTP. Afin de compiler cette application ainsi que toutes ses dépendances et obtenir un fichier possible à déployer sur le serveur *GlassFish*, on se sert du moteur de production *Gradle** (cf. Annexe 2).

3.1.2 Mise à jour des fonctionnalités

Afin de créer un site Web dynamique et sécurisé, le langage choisi est le PHP selon un modèle MVC. Cela permet ainsi de bénéficier d'une arborescence claire et sécurisée côté client. Ce site permet de faire appel de façon simple pour tous aux services *Brunch*.

Au cours du développement du projet et après discussion avec le commanditaire, le cahier des charges a subi quelques petites évolutions. L'utilisateur "standard" n'a donc plus accès à la fonctionnalité "display user", seul un administrateur le peut. De plus, les fonctionnalités "display project" et "display version" sont supprimées (cf. Annexe 3).

D'un autre côté, une gestion de compte et de l'utilisateur est ajoutée comme fonctionnalité. L'utilisateur possède désormais un compte concret avec un mot de passe et peut le changer à loisir.

3.1.3 Base de données

Afin de simplifier la communication entre les données et le site, une base de données MySQL est mise en place. Cette base de données comporte quatre tables : Users, Projects, Versions, Services (cf. Annexe 4).

Chaque table comprend un attribut "id" auto implémenté qui représente sa clef primaire. Tous les attributs d'une table sont préfixés par la première lettre du nom de la table.

Arnaud SOULIER

La table "Users" contient un attribut "u_login" pour identifier l'utilisateur, "u_password" pour stocker son mot de passe, chiffré avec une clef de hachage, et "u_admin" pour définir s'il s'agit d'un compte administrateur ou non. L'attribut "u_login" appartient à un index unique afin d'empêcher deux utilisateurs d'avoir le même identifiant.

La table "Projects" contient un attribut "p_name" pour nommer le projet et "u_id", une clef étrangère, pour définir à quel utilisateur ce projet est rattaché. Les attributs "p_name" et "u_id" appartiennent à un index unique afin d'empêcher un même utilisateur d'avoir deux projets du même nom. En revanche, il peut y avoir deux projets au nom identique dans la base de données si les utilisateurs référencés sont différents.

La table "Versions" est identique à la table "Projects". Elle comporte les attributs "v_name" et "p_id". Ce dernier étant une clef étrangère et les deux attributs sont regroupés dans un index Unique.

La table "Services" contient un attribut "s_name", le nom du service, "s_description" qui indiquera globalement l'utilité du service et "s_url" qui représente quasiment le nom du service mais sans caractères spéciaux (espaces, majuscules, etc...). Ce dernier appartient à un index unique et permet de remplir l'URL faisant appel à *Brunch*.

(cf. Annexe 5 pour la création des tables en détail)

3.2 Résultats et perspectives

3.2.1 Interfaces utilisateur

L'interface utilisateur se présente sous la forme d'un site Web. Il obéit à l'architecture MVC (cf. Annexe 6). Il contient donc un contrôleur qui fait appel à une vue principale avec en paramètre la page à afficher. Cette vue génère un en-tête, un pied de page ainsi que le corps de la page composée de la vue passée en paramètre. Chaque vue est décomposée en un ou deux fichiers. Le premier affiche une page HTML avec un formulaire et le second traite les données apportées par le formulaire. Dans le cas où il n'y a qu'un fichier, celui-ci traite directement les données sans avoir besoin de formulaire.

3.2.2 Perspective

De façon générale, tous les éléments primordiaux du cahier des charges sont présents et respectés. Seule la fonctionnalité d'affichage des données ("Display projects/versions") n'est pas développée mais le cahier des charges ayant changé en cours de route, ce n'est plus nécessaire.

Pour l'instant la table "Versions" de la base de données n'est pas parfaitement exploitée. Elle pourrait être améliorée afin de permettre une meilleure gestion sur le site durant l'appel des services d'analyses.

Déploiement des services d'analyse de qualité logiciel vers les serveurs de Polytechnique
Montréal et développement d'une interface Web utilisateur intuitive

Arnaud SOULIER

D'un point de vu design, le site peut grandement être amélioré. L'apparence n'étant pas un besoin fonctionnel du cahier des charges, elle n'a pas été approfondie. On peut notamment lui ajouter un caractère adaptatif, qui n'est que partiellement développé.

4 Manuel d'utilisation

4.1 Gestion des utilisateurs

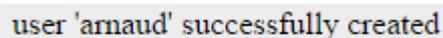
4.1.1 Création d'un compte utilisateur



The image shows a web form titled "Create User". It contains two input fields: "User name *" and "Password *". Below these is a reCAPTCHA widget with the text "Je ne suis pas un robot" and the reCAPTCHA logo. At the bottom of the form are two buttons: "Submit" and "Reset".

Figure 4 - Formulaire création utilisateur

Avant de pouvoir utiliser les services d'analyse de *Brunch*, il vous faudra procéder à la création d'un compte. Pour ce faire, cliquez simplement sur le bouton "Register" en haut à droite. Vous arrivez alors sur une page vous demandant de renseigner un nom d'utilisateur ainsi qu'un mot de passe. Vous devrez également valider un CAPTCHA. Les CAPTCHA vous seront également demandés à chaque étape de suppression. Une fois rentrés et validés, vous serez automatiquement connecté sur votre nouveau compte et aurez ainsi accès à toutes les fonctionnalités de *Brunch*. Si le nom d'utilisateur que vous avez choisi est déjà pris, le site vous l'indiquera et vous demandera d'en choisir un autre.



user 'arnaud' successfully created

Figure 5 - Affichage création utilisateur

4.1.2 Connexion

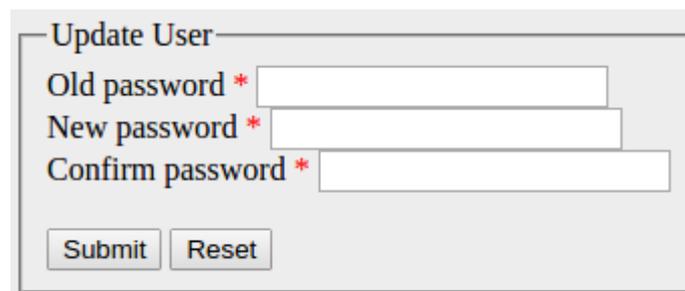


A login form with a light gray background. It contains two text input fields: "User name *" and "Password *". Below the fields is a button labeled "Login".

Figure 6 - Formulaire connexion

Si vous possédez déjà un compte sur le site, il vous suffira de vous connecter via le bouton "Login" en haut à droite. Il vous sera alors demandé de rentrer vos identifiants pour accéder au reste du site.

4.1.3 Modification de vos paramètres et suppression du compte



An "Update User" form with a light gray background. It contains three text input fields: "Old password *", "New password *", and "Confirm password *". Below the fields are two buttons: "Submit" and "Reset".

Figure 7 - Formulaire mise à jour mot de passe

Une fois connecté, vous aurez accès aux paramètres de votre compte depuis l'onglet "User". Vous pourrez ici modifier votre mot de passe dans la partie gauche de la page. Il vous sera alors demandé votre mot de passe actuel ainsi que le nouveau mot de passe désiré, accompagné d'une confirmation.

password of 'arnaud' successfully changed

Figure 8 - Affichage mise à jour mot de passe

Arnaud SOULIER



Figure 9 - Formulaire suppression utilisateur

Vous pourrez également décider de supprimer votre compte. Tous vos projets et versions enregistrés jusqu'à maintenant seront alors effacés et vous serez déconnecté du site.

user 'arnaud' successfully deleted

Figure 10 - Affichage suppression utilisateur

4.2 Gestion des projets

Pour commencer à travailler avec *Brunch*, rendez vous sur l'onglet "Project". La nouvelle page affichée vous offrira alors le choix entre créer un nouveau projet ou supprimer un ancien projet.

4.2.1 Création d'un projet

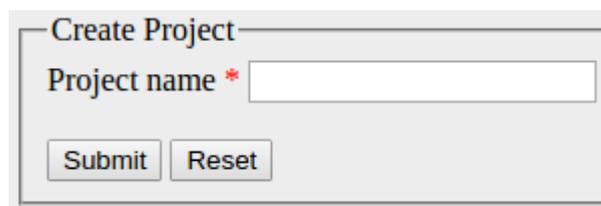


Figure 11 - Formulaire création projet

Dans la partie gauche, vous pourrez renseigner le nom du projet que vous souhaitez créer. Si vous possédez déjà un projet du même nom, il vous sera demandé d'en saisir un autre. Un fois votre projet créé avec succès vous serez automatiquement redirigé vers la page de gestion des versions.

project 'project1' successfully created
[Next step ->](#)

Figure 12 - Affichage création projet

4.2.2 Suppression d'un projet



The screenshot shows a web form titled "Delete Project". At the top, there is a dropdown menu labeled "Project *" with "project1" selected. Below this is a reCAPTCHA challenge area containing a checkbox, the text "Je ne suis pas un robot", and the reCAPTCHA logo with the text "Confidentialité - Conditions". At the bottom of the form are two buttons: "Submit" and "Reset".

Figure 13 - Formulaire suppression projet

Dans la partie droite de la page, vous pourrez demander la suppression d'un de vos projets. Pour cela il vous faudra renseigner quel projet vous souhaitez supprimer et valider le CAPTCHA. Toutes les versions ajoutées à ce projet seront alors effacées.

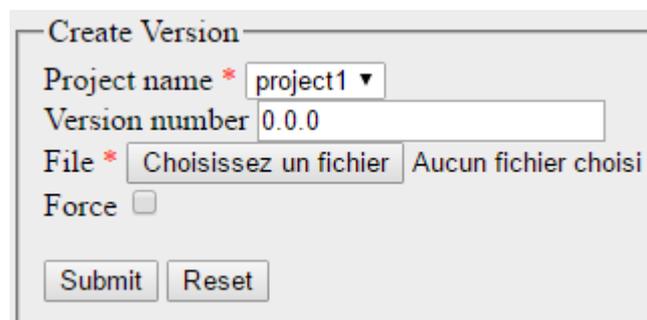
project 'project1' successfully deleted

Figure 14 - Affichage suppression projet

4.3 Gestion des versions

De la même façon que pour les projets, vous aurez la possibilité d'ajouter ou de supprimer une version d'un de vos projets. Pour cela, rendez-vous à l'onglet *Versions*.

4.3.1 Ajout d'une version



The screenshot shows a web form titled "Create Version". It contains several input fields: "Project name *" is a dropdown menu with "project1" selected; "Version number" is a text input field containing "0.0.0"; "File *" is a file selection button with the text "Choisissez un fichier" and "Aucun fichier choisi"; and "Force" is a checkbox. At the bottom of the form are two buttons: "Submit" and "Reset".

Figure 15 - Formulaire création version

Un nouveau champ apparaît ici à la différence de la création de projets. Il s'agit du projet auquel vous souhaitez raccrocher cette version. Vous devrez également renseigner le numéro

Arnaud SOULIER

de la version au format "x.x.x". Si un paramètre n'est pas valide, vous en serez informé et vous pourrez le modifier. Une fois la version créée avec succès, vous arrivez sur la page d'appel des services de *Brunch*.

version '0.0.0' of 'project1' successfully added
[Next step ->](#)

Figure 16 - Affichage création version

4.3.2 Suppression d'une version



Delete Version

Project * project1 ▼

Version number *

Je ne suis pas un robot

reCAPTCHA
Confidentialité - Conditions

Submit Reset

Figure 17 - Formulaire suppression version

Lorsque que vous n'aurez plus besoin d'une version de votre projet, il vous suffira de la supprimer dans la partie droite de l'onglet "Versions". Il vous faudra renseigner le projet auquel appartient cette version ainsi que le numéro de la version. Encore une fois, un CAPTCHA est présent pour les suppressions.

version '0.0.0' of 'project1' successfully deleted

Figure 18 - Affichage suppression version

4.5 Gestion administrateur

Lorsque l'utilisateur connecté est un administrateur, il a accès à des fonctionnalités supplémentaires. Il est en mesure d'afficher la liste de tous les utilisateurs enregistrés.

5 Rapport d'activité

Durant tout mon stage, j'ai suivi une méthode de gestion de projet basé sur les méthodes agiles. Je disposais de trois tâches directrices que l'on pourrait définir comme mes itérations. J'ai divisé ces trois itérations en plusieurs petites étapes. Je ne me suis pas fixé de dates précises pour ces différentes étapes ni pour les trois grandes itérations car j'ai utilisé une méthode de mitigation des risque par "tampon"* pour la gestion des risques. J'avais ainsi une marge de manœuvre très intéressante en cas de problème. Afin de définir les petites étapes, je prenais le temps chaque lundi de décider ce que je devais faire durant la semaine et de commencer à anticiper les tâches à accomplir pour la semaine suivante. Ces dernières étaient confirmées ou reportées en début de semaine suivante. De plus, je réalisais un compte rendu oral ou écrit à mon tuteur chaque fin de semaine pour le tenir informé de mon avancée, des difficultés rencontrées ou des questions que je me posais.

5.1 Découverte

Durant les trois premières semaines de mon stage, je me suis plongé dans le code et les documents que l'on m'avait fournis à mon arrivée. Mon tuteur étant absent lors de la première semaine pour raison professionnelle à Ottawa, afin de participer à un groupe d'évaluation des demandes des subventions d'un programme fédéral, je me suis documenté sur les outils utilisés et imposés. Je devais reprendre un travail déjà commencé, il me fallait donc comprendre ce travail avant de pouvoir le continuer. Étant donné que je n'avais pas encore de notions dans le langage et les outils utilisé, il m'a fallu du temps pour apprendre les bases et pouvoir m'en servir.

Dès que j'ai pu, j'ai réalisé une rencontre avec mon tuteur pour voir avec lui les détails de ma mission. Nous nous sommes mis d'accord sur un plan et les tâches à accomplir (cf. Annexe 7). Nous avons donc établi un cahier des charges suivant ces quelques tâches majeures. Après cela, j'ai continué mes recherches afin d'être au point avec les technologies que je devais employer. J'ai également eu accès au rapport des précédents stagiaires.

5.2 Faire marcher le travail d'avant

La première tâche principale à exécuter était de faire fonctionner le code de mes prédécesseurs. J'ai donc commencé par récupérer la totalité de leur travail.

Avant toute chose, je me suis attaqué au problème de la compilation. Ayant réussi à comprendre comment utiliser *Gradle*, j'ai tenté de compiler le projet mais de nombreuses erreurs sont apparues. Sachant que mes prédécesseurs avaient travaillé sur différents OS, l'architecture du projet n'était pas cohérente avec le code que j'avais récupérer. Après avoir restructuré tout le projet, et débogué quelques lignes incomplètes dans le code, j'ai pu accomplir une compilation.

Arnaud SOULIER

Une fois le projet correctement compilé, il a fallu installer et configurer le serveur *GlassFish* afin de pouvoir y déployer l'application. Je me suis encore une fois inspiré des données que j'avais récupérées de mes prédécesseurs.

Le serveur était accessible, l'application était compilé, mais elle était encore incomplète. Je me suis donc employé à finir le code pour enfin pouvoir commencer la réalisation du site Web.

5.3 Site Web

Une fois que *Brunch* fut opérationnel et déployé, j'ai commencé la création du site Web en PHP selon le modèle MVC. Il n'y avait cependant à ce moment-là aucune base de données ni aucun modèle. Seul le contrôleur et les vues étaient présents à ce stade du développement.

J'ai peu à peu ajouté les fonctionnalités du site. Dans un premier temps j'ai construit tous les formulaires récupérant les données nécessaires à la création ou la suppression d'utilisateur, de projet ou de version ainsi que l'appel des services. Il m'a ensuite fallu réussir à accéder à *Brunch*, qui nécessitait une authentification dans l'en-tête de la requête HTTP depuis le PHP. Cette démarche m'a pris un temps non négligeable. Cependant, grâce à la méthode de mitigation des risques par tampon, j'ai pu adapter mon temps de travail pour pallier au problème. Dans un second temps j'ai ajouté une fonction de connexion et d'identification pour faciliter l'accès et l'utilisation du site Web avant de me lancer dans la base de données.

5.4 Base de données

À ce stade-là, l'application n'était déployée que localement. J'ai donc effectué une rencontre avec les responsables du réseau informatique de l'École afin de pouvoir créer et avoir accès à une instance sur l'un des serveurs de l'École. Une fois tout ceci terminé, j'ai commencé l'installation et la création d'une base de données MySQL. Cette étape du projet n'a pas vraiment posé de problèmes.

5.5 Finition

Enfin durant les deux dernières semaines du stage, je me suis consacré à de petites finitions.

La première fut de débloquer, avec l'aide des responsables du réseau informatique, l'accès au port 80 de mon instance sur le serveur. Une fois fait, le site pouvait être accessible sur un navigateur uniquement par son adresse, sans avoir à spécifier de port.

En parallèle, j'ai commencé à mettre en place le CSS du site afin d'obtenir un site Web présentable, même si ça ne faisait pas partie des besoins immédiats de mon tuteur.

Déploiement des services d'analyse de qualité logiciel vers les serveurs de Polytechnique
Montréal et développement d'une interface Web utilisateur intuitive

Arnaud SOULIER

Enfin, je me suis occupé de la rédaction de mon rapport de stage, en le faisant régulièrement consulter par mon tuteur pour avoir son avis.

Arnaud SOULIER

Conclusion

Ce projet visait à développer et déployer une interface utilisateur pour les services d'analyse de qualité logiciel *Brunch*. Au vu des résultats obtenus, l'objectif est correctement rempli. Étant confié à de nouveaux stagiaires régulièrement, il évoluera très vite vers sa nouvelle version.

Ce stage m'a permis d'acquérir de nouvelles connaissances sur les outils *GlassFish*, *Gradle* et sur le Java EE qui pourront certainement me resservir dans la suite de mes études ou dans mes futurs projets professionnels. J'ai également pu consolider et appliquer mes connaissances en PHP déjà acquises durant mes études. De plus, j'ai pu expérimenter une nouvelle approche de travail en autonomie.

Pour finir je dirai que cette expérience fut très enrichissante, autant d'un point de vue technique qu'humain. C'était également une grande opportunité de voyager et un immense plaisir de découvrir une nouvelle approche culturelle ainsi qu'un éventail de plats typiques à savourer.



Références bibliographiques et sitographiques

Ouvrages consultés

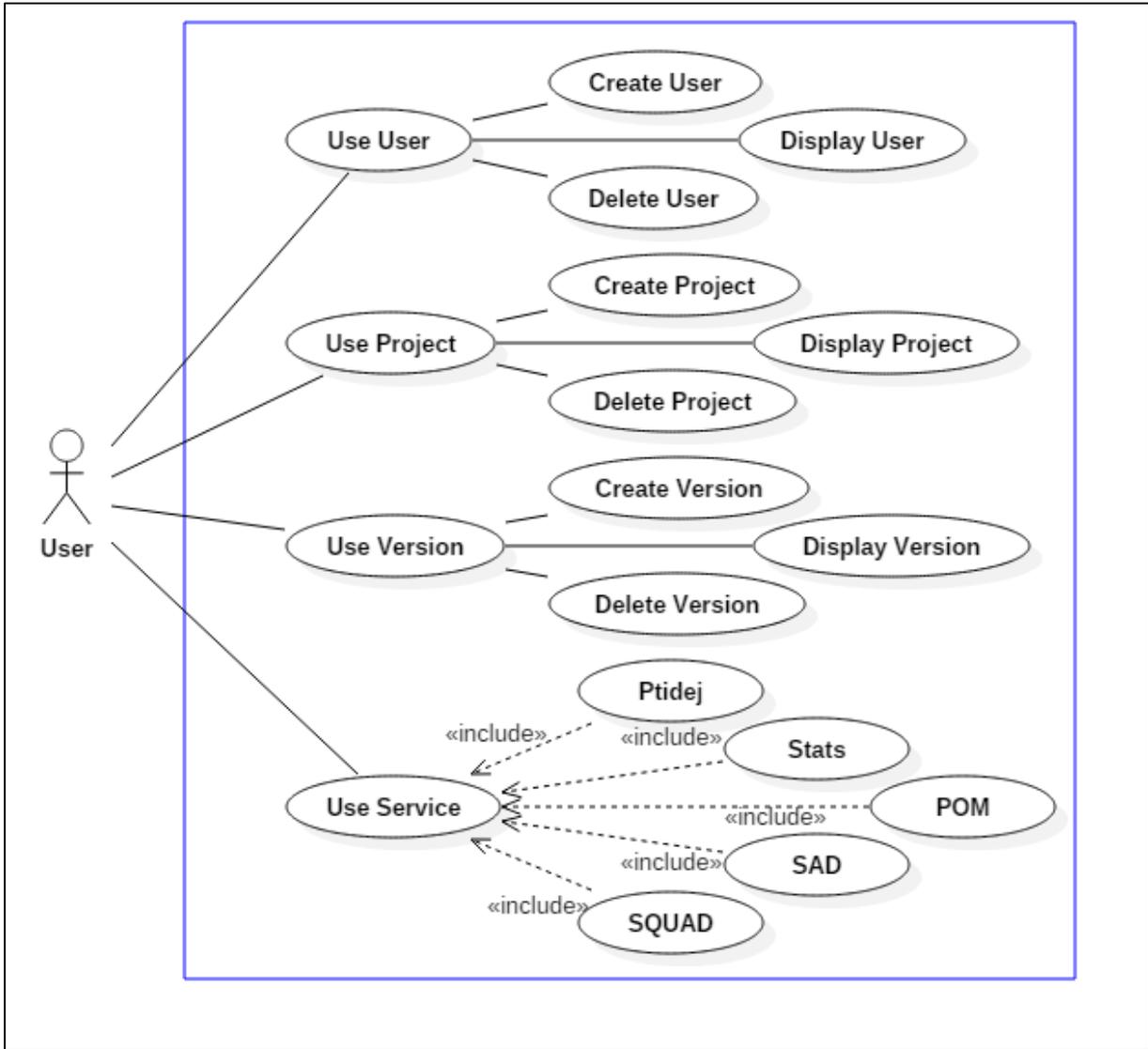
- Web Design in a Nutshell, 2nd Edition by Jennifer Niederst

Site internet consultés

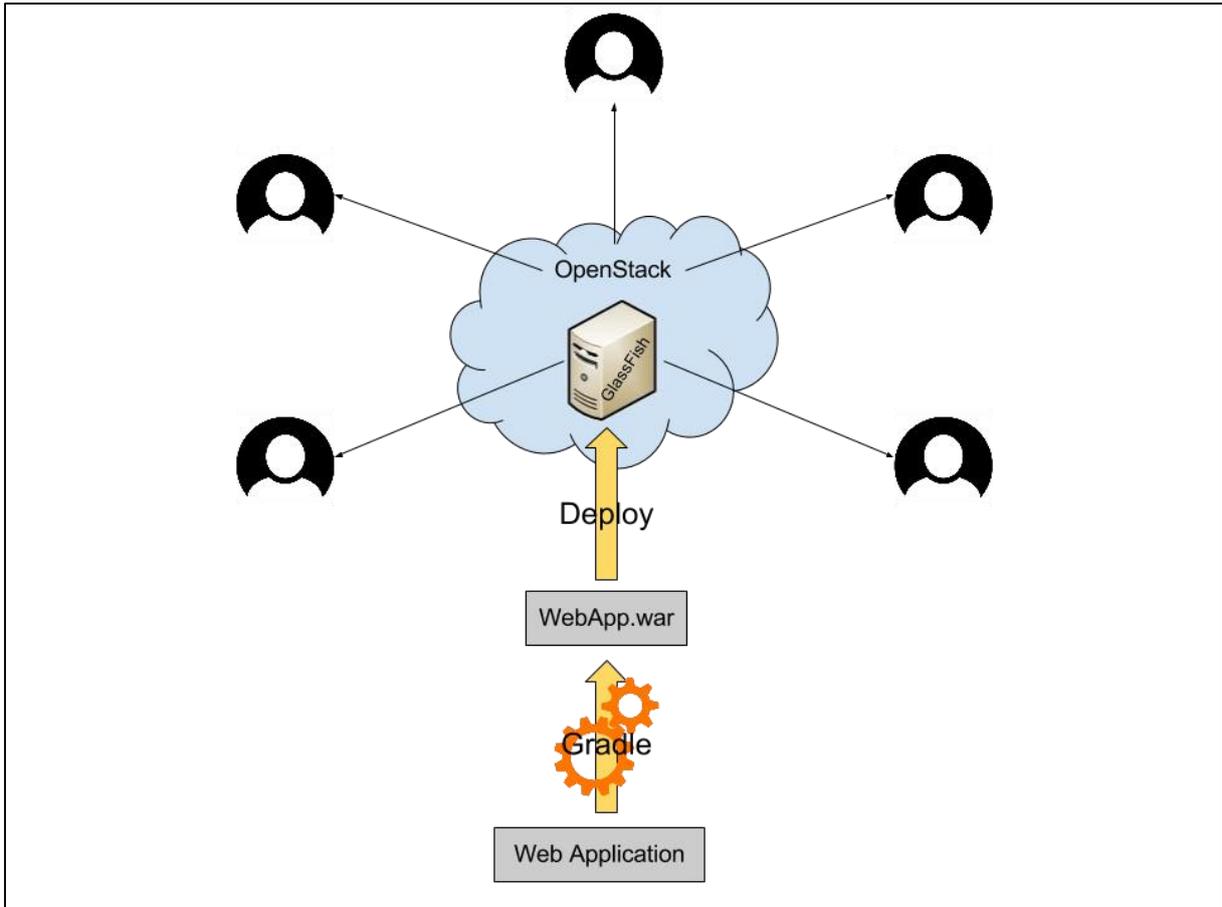
Adresse de la page	Type de site	Information recherchée
https://docs.oracle.com/	Documentation	Javadoc de fonctions et class
php.net	Documentation	Documentation des fonctions php
https://openclassrooms.com/forum/	Forum	Solutions pour corriger des bugs ou idée de méthodes.

Annexes techniques

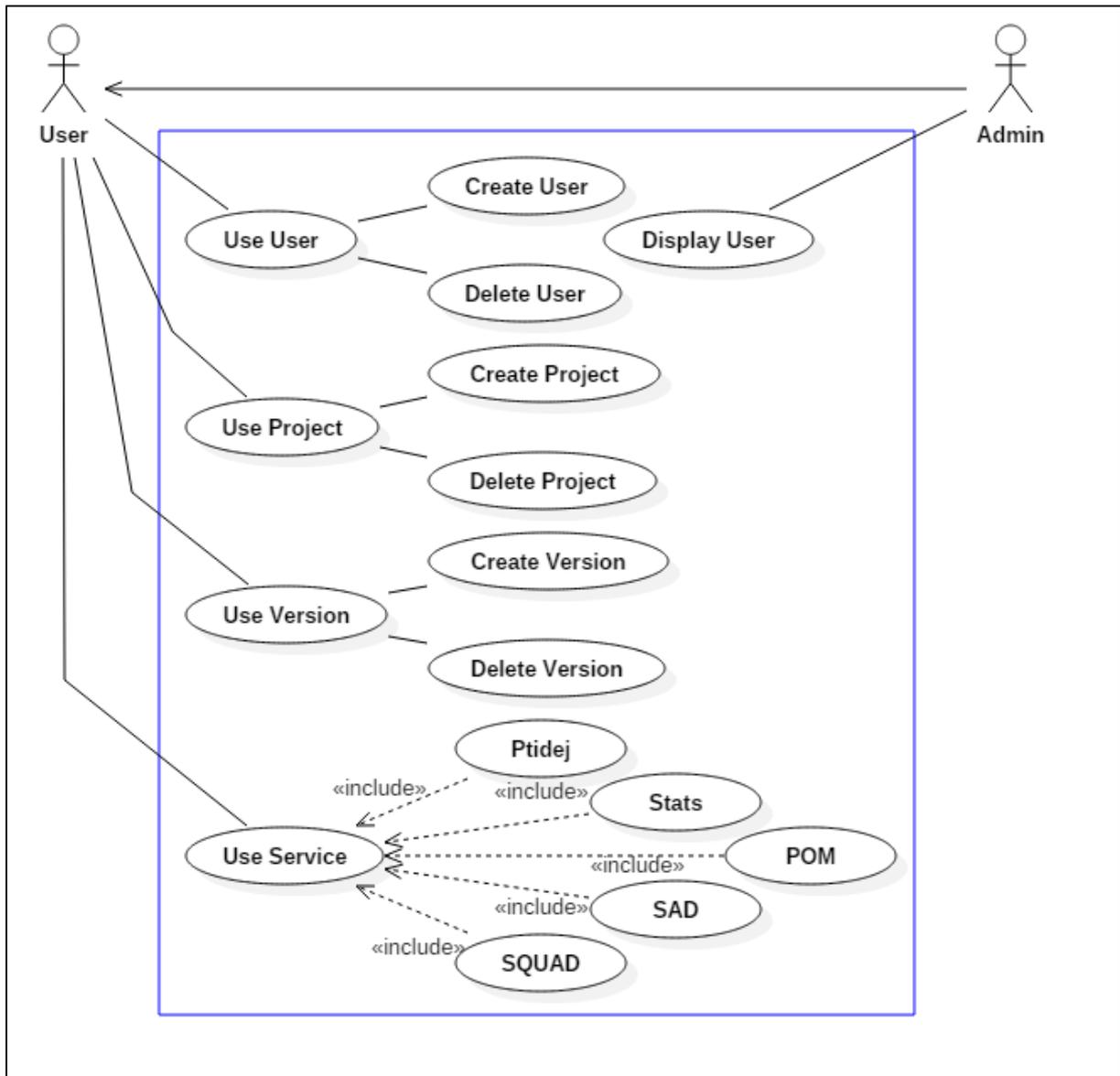
Annexe 1 - Premier cas d'utilisation	III
Annexe 2 - Schéma déploiement application	IV
Annexe 3 - Second cas d'utilisation	V
Annexe 4 - Schéma base de données	VI
Annexe 5 - Script de création de la base de données	VII
Annexe 6 - Architecture MVC	VIII
Annexe 7 - Brouillon réunion avec tuteur	IX



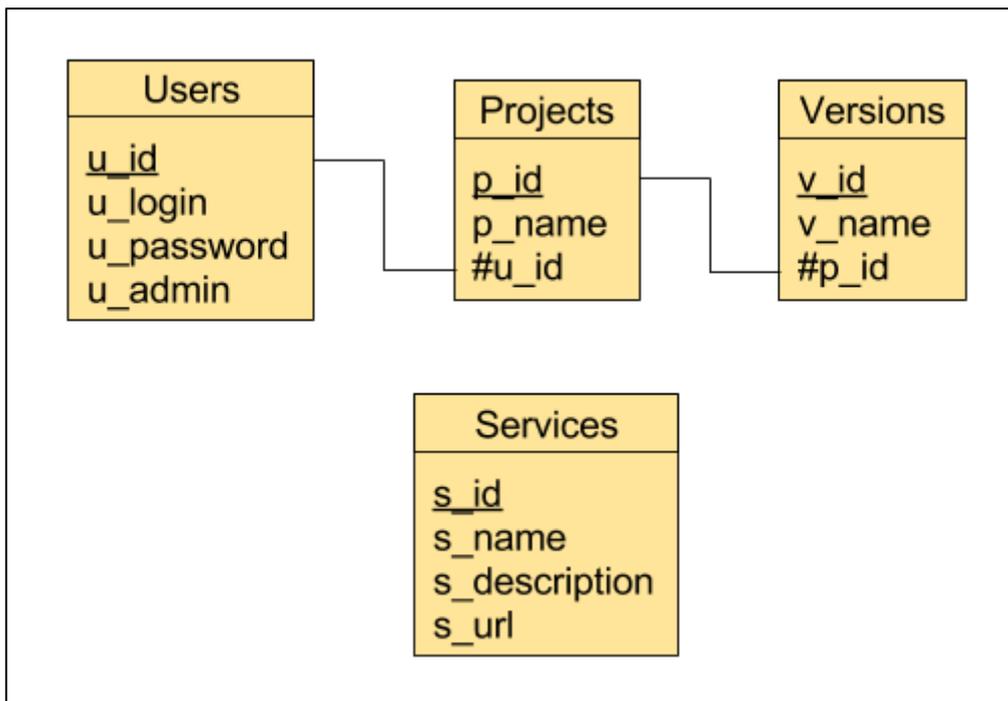
Annexe 1 - Premier cas d'utilisation



Annexe 2 - Schéma déploiement application



Annexe 3 - Second cas d'utilisation



Annexe 4 - Schéma base de données

```

CREATE TABLE Users(
    u_id int NOT NULL AUTO_IMPLEMENT,
    u_login varchar(255) NOT NULL,
    u_password varchar(255) NOT NULL,
    u_admin bool NOT NULL,
    CONSTRAINT pk_users PRIMARY KEY(u_id),
    CONSTRAINT uq_users UNIQUE(u_login));

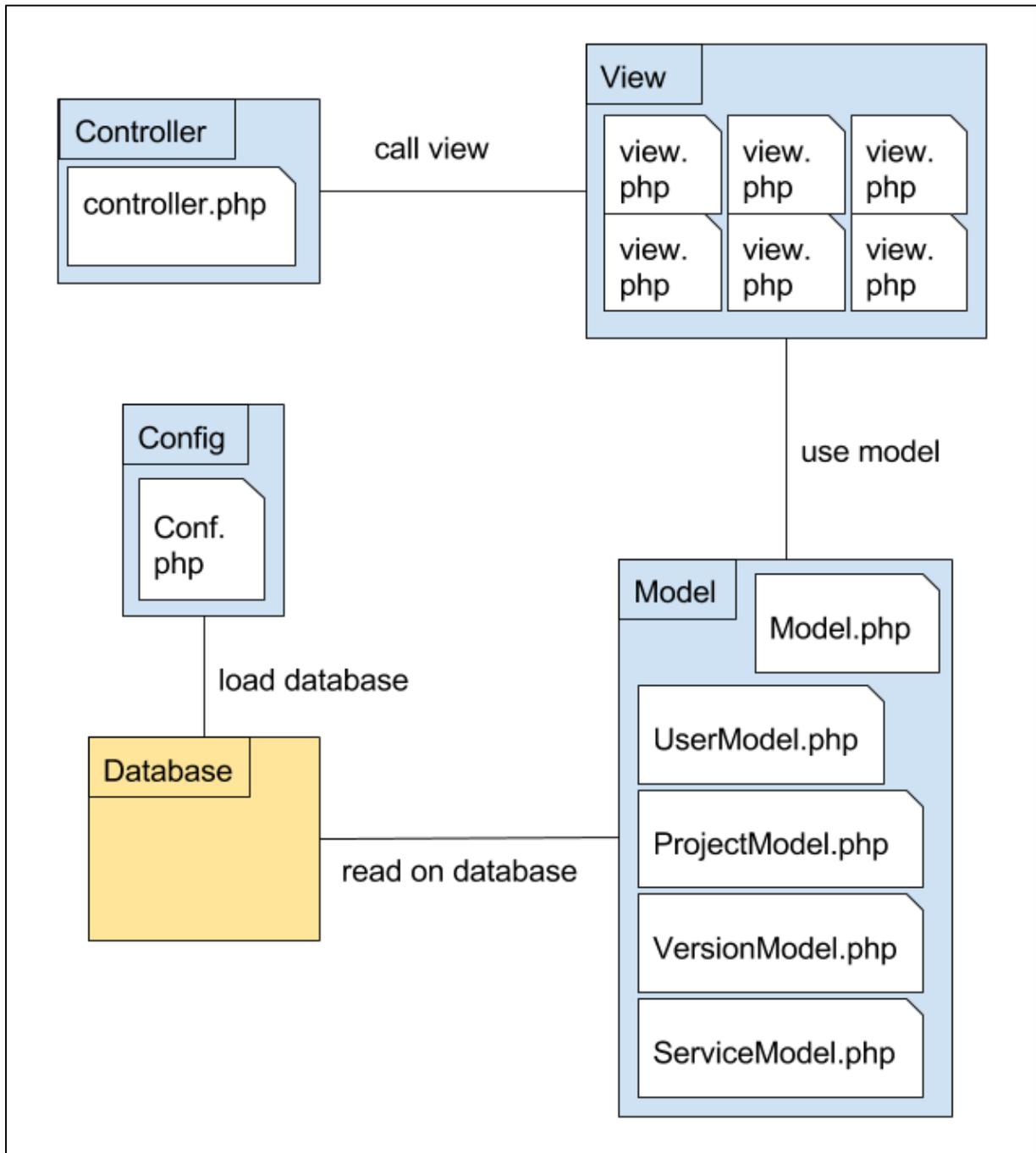
CREATE TABLE Projects(
    p_id int NOT NULL AUTO_IMPLEMENT,
    p_name varchar(255) NOT NULL,
    u_id int NOT NULL,
    CONSTRAINT pk_projects PRIMARY KEY(p_id),
    CONSTRAINT fk_projects FOREIGN KEY(u_id) REFERENCE Users(u_id),
    CONSTRAINT uq_projects UNIQUE(p_name, u_id));

CREATE TABLE Versions(
    v_id int NOT NULL AUTO_IMPLEMENT,
    v_name varchar(255) NOT NULL,
    p_id int NOT NULL,
    CONSTRAINT pk_versions PRIMARY KEY(v_id),
    CONSTRAINT fk_versions FOREIGN KEY(p_id) REFERENCE Projects(p_id),
    CONSTRAINT uq_versions UNIQUE(v_name, p_id));

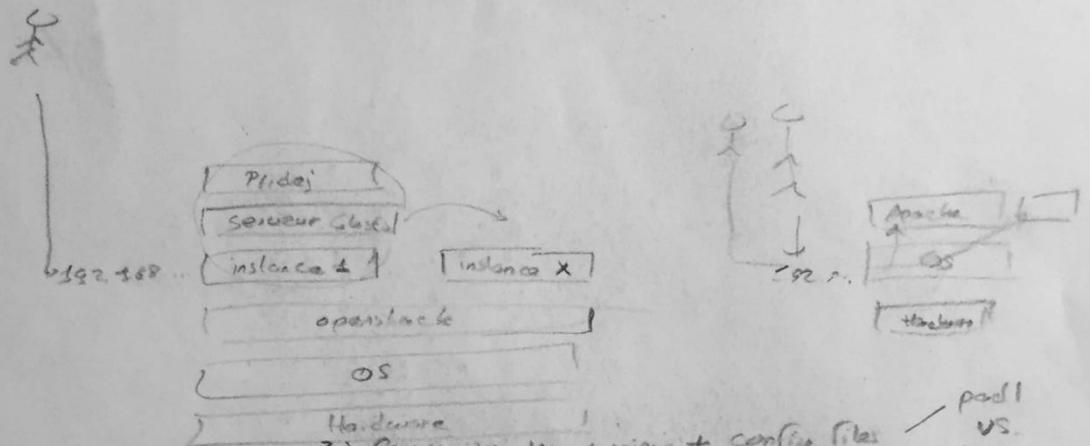
CREATE TABLE Services(
    s_id int NOT NULL AUTO_IMPLEMENT,
    s_name varchar(255) NOT NULL,
    s_description varchar(255) NOT NULL,
    s_url varchar(255) NOT NULL,
    CONSTRAINT pk_services PRIMARY KEY(s_id),
    CONSTRAINT uq_services UNIQUE(s_url));

```

Annexe 5 - Script de création de la base de données



Annexe 6 - Architecture MVC



- 3) Réorganiser les dossiers + config files
 - 2) Recompilation du WAR
 - 1) Comprendre la création des usages
 - 0) Comment openstack déploie les instances
- padl vs PADL (répondre error 404)

- 1) probe it avec
 - 2) Services
 - 3) UI
- } ←

5) Fusionner "Branch" "Pridej"

Annexe 7 - Brouillon réunion avec tuteur

4^e de couverture

Ptidej (Pattern Trace Identification, Detection, and Enhancement in Java) est une suite d'outils offrant des services d'analyse de qualité logiciel à l'aide de métriques et de patrons. *Brunch* est une version Web des services de *Ptidej*. C'est une application Web développée en Java EE et accessible depuis un serveur de l'École Polytechnique de Montréal par le biais d'un site Web développé en PHP à l'adresse brunch.soccerlab.polymtl.ca. Ce projet a été commencé par plusieurs stagiaires et continuera d'être développé par de nombreux autres.

Mots clefs : application Web, serveur, déploiement, Java, PHP services, qualité, patron, métrique.

Ptidej (Pattern Trace Identification, Detection, and Enhancement in Java) is a tool suite providing software quality analysis services using metrics and patterns. *Brunch* is a Web version of the *Ptidej* services. This is a Web application developed in Java EE and reachable from a Polytechnic School of Montréal's server through a Website developed in PHP at brunch.soccerlab.polymtl.ca. This project was initiated by many trainees and should continue to be developed by many others.

Keywords : Web application, server, deployment, Java, PHP, services, quality, pattern, metric.