Élaboration d'une solution de portail d'application d'entreprise

Vincent Dillenschneider

le cnam

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

CENTRE RÉGIONAL DE LORRAINE

MÉMOIRE EN INFORMATIQUE

OPTION SYSTÈMES D'INFORMATION (ISI)

Elaboration d'une solution de portail d'application d'entreprise

---

Présenté par

Vincent DILLENSCHNEIDER

Soutenu le 4 Juillet 2016

JURY

Président du Jury :

Jean-Pierre ARNAUD	CNAM - Professeur Titulaire de la chaire de Réseaux

Membres du Jury :

Michel IANOTTO	CentraleSupélec, tuteur du mémoire
Jean-Marc THOURON	Ministère de l'éducation Nationale / Lycée de Pont à Mousson
Jean-Philippe AUZELLE	Université de Lorraine / ENSEM
Benoit MARCHAL	Université de Lorraine / Direction du Numérique
Christian DUCLOU	Université de Lorraine / Direction du Numérique

Remerciements

Je tiens avant tout à remercier mon entreprise FERCO et son président M. Fabien Schmitz de m'avoir permis de réaliser ce projet.

Je tiens plus particulièrement à remercier M. Laurent Imm, directeur informatique de la société, ainsi que l'ensemble des collaborateurs du service informatique de l'entreprise pour le soutien apporté dans le déroulement de ce sujet.

Je remercie également l'ensemble des intervenants du CNAM et surtout M. Michel Ianotto et M. Christian Duclou pour leurs conseils et le suivi de mon mémoire.

Enfin, je voudrais remercier et féliciter Nicolas Alvarez pour le travail réalisé au sein de ce projet qui s'est déroulé dans le cadre de son apprentissage de Master informatique et que j'ai eu le plaisir d'encadrer pendant deux ans.

Table des matières

1 Introduction

Titulaire d'une licence professionnelle en informatique de concepteur développeur en environnement distribué obtenue à l'université de Strasbourg en 2006, je suis ensuite devenu un auditeur du conservatoire national des arts et métiers (CNAM) depuis le mois de septembre 2009 afin de valider un cursus d'ingénieur avec comme spécialité les systèmes d'information (CYC12). Le choix de ce cursus était une démarche cohérente dans mon apprentissage, je voulais en effet trouver une formation permettant d'affiner la technique tout en délivrant des connaissances supplémentaires sur la gestion d'un système d'information, le management d'une entreprise et les sciences humaines.

Lors de mon apprentissage au CNAM, j'ai pu prendre plaisir à appliquer dans le cadre de mon activité professionnelle des notions que j'ai acquises durant ces cours du soir, cela m'a permis d'apporter des réponses innovantes et adaptées aux problèmes de mon entreprise. Avec ce recul et de la mise en pratique par des cas concrets depuis ces six dernières années, je peux confirmer avoir été très satisfait des cours délivrés par le CNAM.

Je suis maintenant heureux de concrétiser l'achèvement de mon cursus par la rédaction de ce mémoire en présentant un sujet qui a été intéressant à mener et qui présente une valeur ajoutée à mon entreprise. Ce sujet est la mise en place d'un nouveau portail d'application d'entreprise (PAE) ou (portail applicatif), c'est à dire un portail proposant des applications Web qui offrent des services dédiés aux salariés de l'entreprise. Ce nouveau PAE vise à remplacer celui existant (nommé portail Intranet) et à améliorer la qualité des services fournis aux utilisateurs de l'entreprise.

Le mémoire va traiter le sujet à travers cinq parties. La première partie est consacrée à la présentation du contexte du projet en introduisant mon entreprise ainsi que le système d'information existant. Une description fonctionnelle de l'ancien portail applicatif nommé "portail Intranet" clos cette première partie. La deuxième partie va permettre de détailler le cahier des charges en effectuant une analyse plus complète du portail Intranet et en décrivant l'étude d'une nouvelle solution. La troisième partie est dédiée à la conception et réalisation du PAE. Dans une quatrième partie, je vais pouvoir décrire la formation du personnel sur ce nouveau PAE, ainsi que sur la communication qui a été effectuée. Enfin, la dernière partie va permettre de dresser un bilan suite à la réalisation et à l'exploitation de ce nouveau PAE.

2 Présentation du contexte et du sujet de l'étude

2.1 Le sujet de l'étude

Le portail Intranet est un PAE développé en 2008 pour la société FERCO et ce dernier n'est plus en adéquation avec les besoins de la société : qualité exigée, productivité, soucis de maintenance et de performances. C'est pourquoi le projet d'élaborer une nouvelle solution de PAE a été décidé.

Le sujet de l'étude est donc de réaliser un nouveau portail applicatif dont le but est de remplacer le portail Intranet existant, tout en améliorant la qualité des services fournis et en supprimant les limitations du système actuel.

2.2 Contexte et environnement

2.2.1 L'entreprise FERCO

La société FERCO a été créée en 1934 et se trouve être une filiale du groupe allemand Gretsch-Unitas (G-U) depuis son rachat en 1977. Elle est située à Sarrebourg en Moselle (figure 2.1) et représente aussi la plus grande entreprise française (et un des plus grands producteur européen) de ferrures et serrures de bâtiments et utilisent différents type de matériaux (aluminium, bois et PVC) pour ses produits.

En 2005, M. Fabien Schmitz devient l'actuel président de FERCO. Le groupe G-U comprend une cinquantaine de sociétés réparties dans trente cinq pays, dont les trois principales sont : FERCO, BKS (technique de portes) et G-U Automatics (systèmes de portes automatiques). Le groupe est dirigé par les frères Michael et Julius Von Resch, qui sont les deux principaux actionnaires.

FIGURE 2.1 – Implantation géographique de Ferco

Le groupe G-U

En 1907, l'ingénieux technicien Viktor Gretsch se lance dans la fabrication d'impostes à tirage, d'arrêts de fenêtres et de petites ferrures. Trois ans plus tard, il fait du commerçant Johann Maus son associé et son gérant. Les descendants de ce dernier, Julius Maus von Resch et Hans Maus, hissent la société au rang d'entreprise d'envergure internationale. Les frères Julius et Michael von Resch sont aujourd'hui la troisième génération à diriger le groupe d'entreprises Gretsch-Unitas.

Au milieu des années 1970, la société se transforme en groupe : tout d'abord par l'acquisition de FERCO, le plus grand fabricant français de ferrures, ensuite par la reprise de la grande marque de serrures BKS et par la création de propres entreprises de services. Le groupe G-U est présent sur les cinq continents (voir figure 2.2). Il est composé de neuf sites de production dont trois en France, trois en Allemagne, un en Chine, un en Turquie, un au Canada et trente-cinq filiales de distributions et agences commerciales. Le groupe G-U compte actuellement un effectif de 3700 personnes à travers le monde.

FIGURE 2.2 – Implantation de Gretsch-Unitas dans le monde

2.2.1.1 Chiffres clés de la société FERCO

Effectif

La société FERCO compte actuellement 813 collaborateurs, la figure 2.3 représente la répartition socioprofessionnelle de l'entreprise.

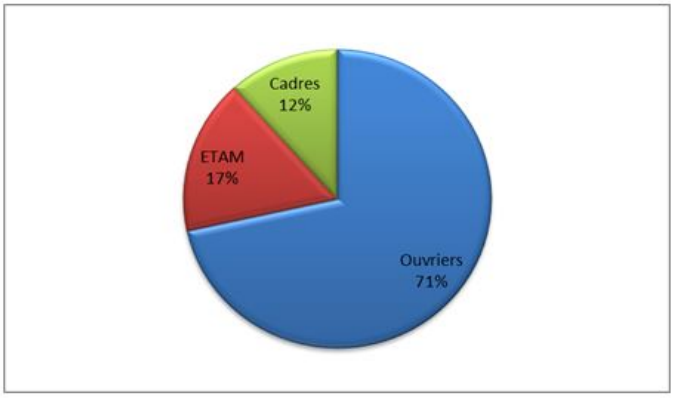


FIGURE 2.3 – Répartition socioprofessionnelle actuelle de l'entreprise

Chiffre d'affaire

Malgré une baisse du CA par rapport à 2010, suite aux difficultés de certains marchés du groupe, FERCO apporte toujours une activité permettant de continuer à être une source d'emploi dans le secteur de Moselle Sud.

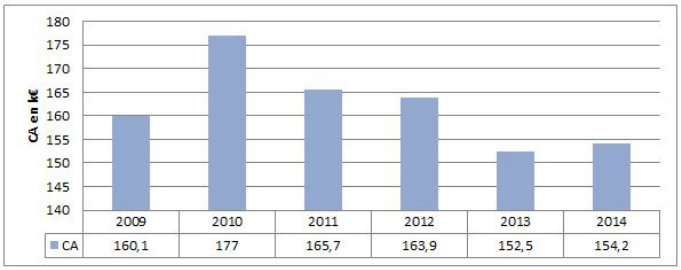


FIGURE 2.4 – Evolution du CA de FERCO en K€

2.2.1.2 Organigramme de l'entreprise FERCO

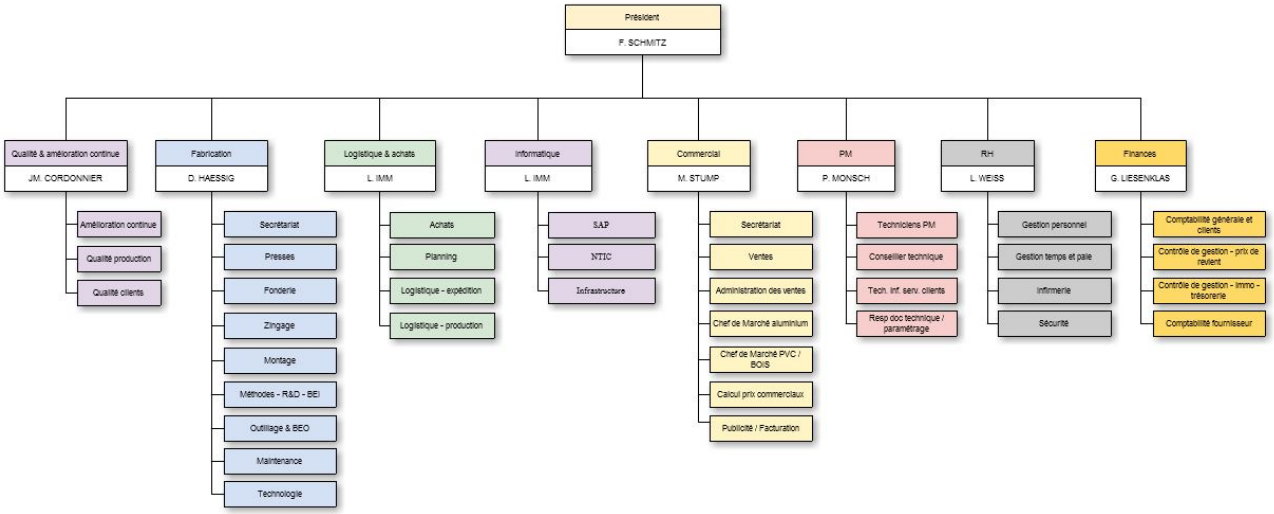


FIGURE 2.5 – Organigramme général de l'entreprise Ferco

2.2.2 Le système d'information

Le système d'information est géré par le service informatique de l'entreprise, qui doit prendre en charge les missions suivantes :

- Maintenir et faire évoluer l'outil informatique (matériel et logiciel) pour assurer un niveau de performance répondant aux besoins à un coût acceptable.
- Assurer aux utilisateurs un service réactif, permettant de résoudre rapidement les problèmes rencontrés et de répondre à leurs besoins ainsi qu'aux évolutions légales ou conventionnelles.
- Conseiller, former et informer les utilisateurs dans tout ce qui du domaine informatique (utilisation SAP et autres logiciels, matériel, sécurité), afin d'augmenter leurs autonomie et leurs permettre de gagner en efficacité.
- Participer aux projets faisant appel à des moyens informatiques, en proposant les solutions les mieux adaptées et en mettant en oeuvre celles-ci avec les services concernés.
- Mettre en place et adapter régulièrement les mécanismes concernant la sécurité informatique et veiller au respect des obligations touchant à l'utilisation des moyens informatiques et à l'accès internet.

2.2.2.1 Ressources humaines

Le service informatique de la société est constitué du directeur informatique et de neuf personnes situées dans trois équipes avec des rôles différents.

L'équipe NTIC

L'équipe de développement NTIC (Nouvelles Technologies de l'Information et de la Communication) est composée de deux personnes et a pour principale mission de prendre en charge la conception et le développement d'applications Web, permettant d'améliorer la communication, le suivi et la gestion de l'information de l'entreprise. Pour répondre à certains besoins, cette équipe peut égale-

ment gérer du développement logiciel de type "client lourd" ou installer et configurer des applications Web tierces.

L'équipe SAP

L'équipe SAP assure le développement et le paramétrage du progiciel SAP (l'ERP (Enterprise Resource Planning) qui est utilisé par la société et le groupe Gretsch-Unitas). Les applications développées sur ce progiciel répondent au besoin métier des utilisateurs des autres services (notamment logistique, ressources humaines, achats et commercial). Quatre personnes sont affectées à cette équipe, toutefois pour des besoins de développement ponctuels, les membres de l'équipe NTIC peuvent intervenir sur certains sujets.

L'équipe infrastructure

L'équipe infrastructure compte trois personnes et ses principales missions sont de fournir aux utilisateurs un équipement informatique (PC, imprimante, ordinateur portable, smartphone, environnement logiciel) adapté à leurs besoins, de former les utilisateurs aux logiciels et outils mis à leur disposition et de gérer les interventions de premier niveau survenus (hot-line). Cette équipe fournit également une expertise et un support sur le matériel nécessaire (serveur et configuration de logiciel) à la réalisation des projets informatiques des équipes de développement (NTIC et SAP).

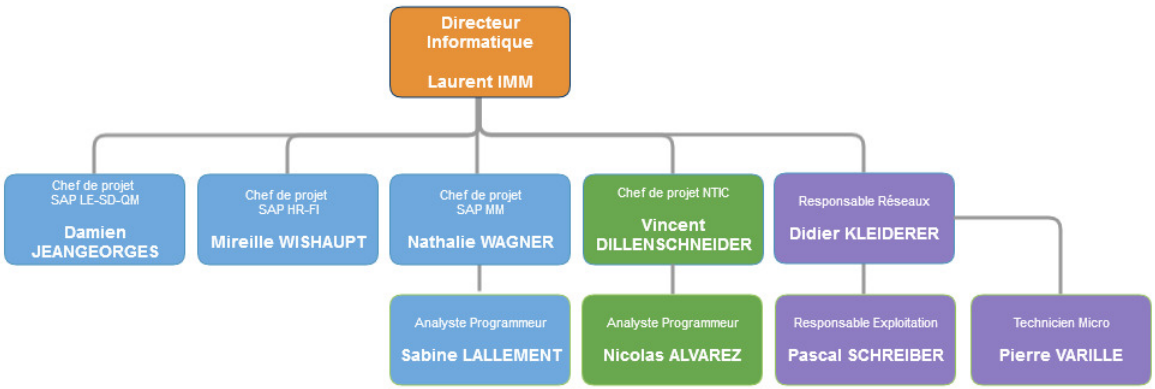


FIGURE 2.6 – Organigramme du service informatique

2.2.2.2 Organisation et management

Gestion des projets et incidents

A FERCO, La gestion des projets informatiques a été revue en 2013, la société utilise maintenant la méthodologie Waterfall (en cascade) qui a l'avantage d'être simple et demande de fournir des livrables à la fin de chaque étape avant de passer à l'étape suivante. Chaque chef de projet utilise cette méthode, ainsi que les standards de documentation pour suivre son projet. Une réunion d'état d'avancement des projets a lieu toutes les deux semaines et est animée par le directeur informatique, ce dernier prend note des informations exprimées par les chefs de projet et décide des actions à entreprendre pour le bon déroulement des projets (budget, priorité, décision stratégique, etc.). Un outil

libre et open source "DotProject" est utilisé comme support lors de ces réunions afin d'avoir un format unique de présentation. Ce logiciel permet également le suivi de la charge de travail du personnel informatique sur les projets et offre également des indicateurs de performances concernant les projets (temps, délais et coût).

La gestion des incidents et des demandes hors projet est réalisée à l'aide du logiciel de gestion libre et open source GLPI (Gestion Libre de Parc Informatique), il permet aux utilisateurs de soumettre des incidents par la création d'un ticket. Lorsque un utilisateur initie un ticket, ce dernier est alors transmis directement au technicien concerné par ce type de demande ou sinon il est envoyé au support de premier niveau (la hot-line) qui l'analyse et prends en charge si possible sa résolution, ou le cas échéant redirige vers la personne la plus compétente sur le domaine afin de pouvoir traiter le problème efficacement. Cet outil permet également de constituer un ensemble d'indicateurs (inspiré des bonnes pratiques de [BAUD, 2012]) : durée moyenne de traitement des tickets, temps de passage, nombre de tickets en retard, etc.

2.2.2.3 Budget

La figure 2.7 montre l'évolution du budget du service informatique, ce dernier est composé d'investissements en matériel et en logiciel. Les coûts liés aux contrats de maintenance et aux licences des logiciels sont également compris dans ce budget. L'écart entre le budget des dernières années et 2013 est expliqué par un projet de renouvellement du parc informatique avec la mise en place de Windows 7 pour remplacer Windows XP sur les postes de travail. Suite à des difficultés économiques de la société, qui trouve une partie de son origine dans le conflit entre la Russie et l'Ukraine (le marché des pays de l'est étant l'un de nos plus grand marché), le budget alloué en 2014 s'inscrit dans le cadre d'une politique de réduction des coûts et se trouve donc être plus faible que les années précédentes.

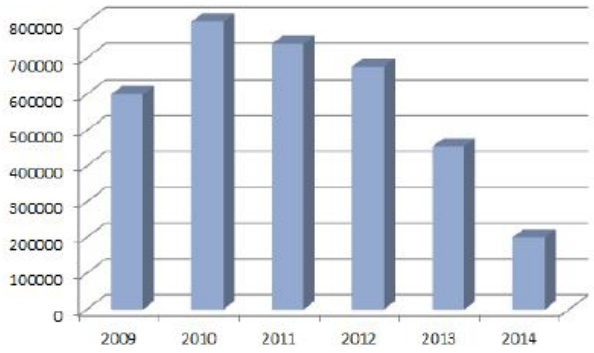


FIGURE 2.7 – Budget du service informatique en euros

2.2.2.4 Architecture technique du système d'information

Une grande partie du système d'information de FERCO est gérée par le progiciel SAP, l'équipe interne de la société effectue uniquement du paramétrage et du développement à distance sur ce système. Physiquement, les serveurs SAP sont centralisés dans les locaux de G-U situés à Ditzingen en Allemagne. La maintenance du système SAP est effectuée par les ingénieurs systèmes de la société Agena (société faisant partie du groupe G-U). Concernant les autres besoins qui ne sont pas couverts par le progiciel SAP (serveurs d'impression, annuaire, serveurs Web, FTP, etc.), ils sont pris en charge par des serveurs virtuels suivis par l'équipe infrastructure de la société FERCO. Pour gérer ses serveurs

virtuels, la société utilise la solution de virtualisation VMWare avec sept serveurs physiques, dont trois sont utilisés pour la gestion des clients légers de l'entreprise.

La figure 2.8 montre une architecture technique simplifiée de notre système d'information (SI).

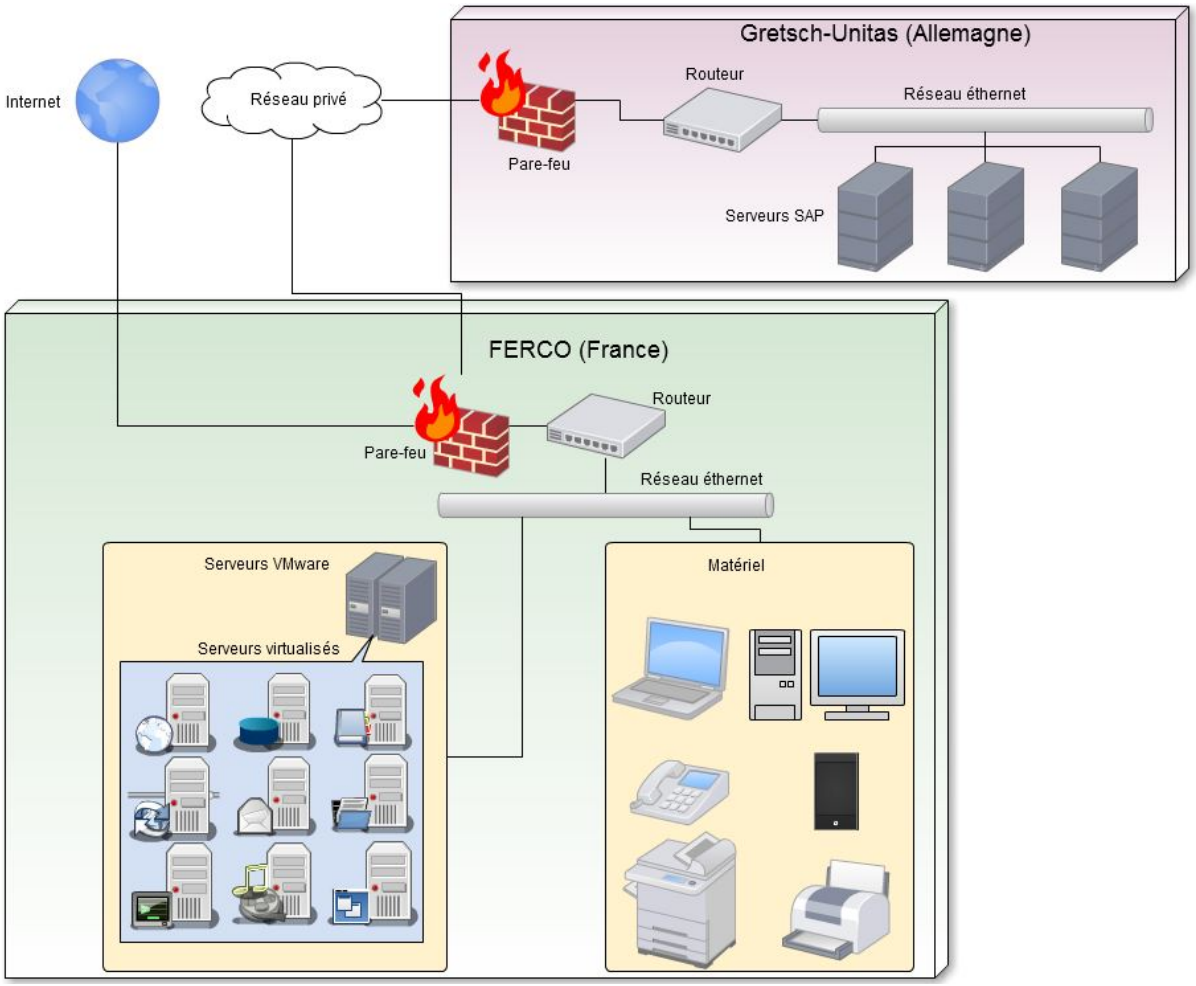


FIGURE 2.8 – Architecture technique simplifié

Les serveurs utilisés par l'équipe NTIC

Pour le développement Web, nous disposons de plusieurs serveurs virtuels hébergés sur des hôtes physiques.

Ces serveurs virtuels ont chacun un rôle particulier :

— fercdeb06 : contient nos dépôts de fichiers sources qui sont gérés par subversion (SVN).
— fercubu06 : il s'agit du serveur Web de production sur lequel se trouve plusieurs applications dont notamment le portail Intranet. Ce serveur contient uniquement les applications développées par l'équipe de développement NTIC.
— fercubu06t : serveur de test, dont l'environnement d'exécution, les applications et leurs dépendances sont une copie parfaite du serveur de production.

— fercubu04 : serveur Web contenant des applications libres et open source qui ne sont pas développées par les développeurs internes de la société. Parmi les applications installées, on peut citer notamment GLPI (solution de gestion des incidents), dotProject (logiciel de gestion de projets) et Piwik (système d'analyse et de statistiques de site web).
— fercubu04 : serveur de test, copie de fercubu04.
— fercubu03 : serveur Web disponible depuis l'extérieur de la société et qui héberge une application Web permettant au médecin du travail de renseigner des données de visites médicales, ainsi qu'une application permettant le transfert de fichiers trop volumineux pour être envoyés par notre système de messagerie entre le personnel de FERCO et des personnes extérieures.
— alfresco01 : serveur contenant le logiciel Alfresco qui est un enterprise content management (ECM), c'est à dire une solution de gestion documentaire permettant de gérer de manière avancée l'ensemble des contenus de l'entreprise.
— alfresco01t : serveur de test, copie de alfresco01.
— fercdeb11 : serveur applicatif contenant le logiciel open source JasperReports Server basé sur Java qui est un outil permettant de réaliser du reporting.
— fercdeb11t : serveur de test, copie de fercdeb11.

2.3 Le portail Intranet

2.3.1 Présentation et historique

Le projet d'établir un site Intranet d'entreprise a été imaginé en 2003 par la direction informatique de l'époque, lors de cette période les besoins exprimés étaient essentiellement des éléments de communication (météo, menu du restaurant d'entreprise, informations sur l'entreprise, etc...). Un prototype de ce site Intranet a été réalisé avec la technologie ASP sur un serveur Microsoft IIS (Internet Information Services), mais a été abandonné en 2005 suite à un plan de restructuration.

En 2007, l'idée de créer un site Intranet pour les salariés de l'entreprise est à nouveau soumise et acceptée par la direction. Suite à cet accord, une première version du site Intranet (écrit avec le langage de programmation PHP) appelée "portail Intranet" a été mise en place en 2008, regroupant les besoins de communication initialement prévues et proposant également une application de gestion des absences pour le personnel (les salariés pouvaient consulter leurs soldes, poser des congés, voir le planning de leurs groupes, etc.). Plusieurs mises à jour ont été apportées à ce portail, certaines incluant de nouvelles applications apportant des services supplémentaires aux salariés (la gestion des plans d'actions par exemple). Parmi ces mises à jours, la plus importante a eu lieue en Novembre 2010 avec la mise en place du framework PHP ZendFramework qui a remplacé le framework "maison" développé par la société.

La figure 2.9 donne un aperçu du portail Intranet actuel.

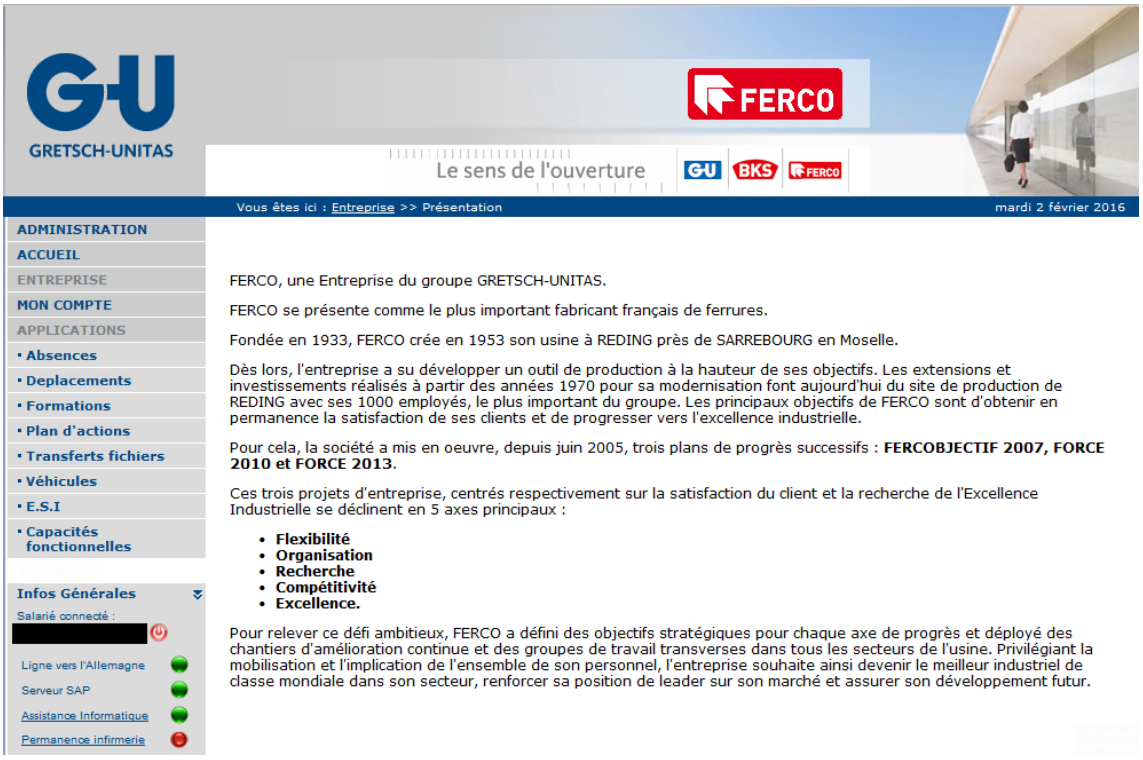


FIGURE 2.9 – Le portail Intranet

2.3.2 Les services proposés

Le portail Intranet est une application Web qui propose différents services aux salariés, on retrouve parmi ces services des applications de communication et de gestion de l'information. Une présentation des principales applications est effectuée ci-dessous.

La gestion des absences

La gestion des absences est un service proposé aux salariés permettant à ces derniers de pouvoir saisir de manière dématérialisée une demande d'absence et de la soumettre à travers un flux de validation aux responsables, une fois ce flux de validation terminé, le système transfère les informations dans le système SAP-HR pour le traitement de la paie. Le salarié peut également consulter le solde de ses congés restants ou encore prendre connaissance du planning de son équipe. Dans cette application, de nombreux indicateurs et reporting sont disponibles à l'attention du service des ressources humaines afin de permettre un meilleur suivi de la gestion des temps des salariés.

La gestion des déplacements

La gestion des déplacements est une application dédiée à la secrétaire de direction qui peut saisir et suivre les déplacements des collaborateurs de la société. Dans cette application, il est possible de déclarer des réservations d'hôtels, de voitures et d'autres types de transport. Le système permet d'éditer un fichier PDF avec les données du déplacement. Une fois le déplacement validé, les données de temps sont transférées dans le système SAP-HR pour le traitement de la paie. Cette application est disponible sur une ancienne version du portail Intranet (v1) et le portail Intranet actuel effectue simplement un lien vers cette application.

L'annuaire de l'entreprise

L'annuaire affiche l'ensemble des informations du personnel de l'entreprise : fonction, photo, secteur, mail, responsable, etc. Les salariés peuvent effectuer des recherches sur cette annuaire en fonction de filtres.

La gestion des formations

Cette application permet au personnel des ressources humaines d'effectuer le suivi des formations de l'entreprise : le gestionnaire effectue la saisie de la formation et le système édite la convocation qui est envoyée aux participants par mail ou papier (si le participant ne dispose pas de messagerie). Une fois la formation effectuée, les informations sont envoyées au système SAP-HR pour le traitement de la paie. En fin d'exercice, un rapport complet est disponible via l'application pour une comparaison avec le budget de formation initialement planifié en début d'année.

Gestion des plans d'actions

L'entreprise a élaboré depuis 20007 une démarche d'amélioration continue sous l'acronyme "PP" (Plan de Progrès), c'est dans ce contexte qu'un outil informatique a été développé dans le portail Intranet afin de faciliter l'implication de l'ensemble des salariés dans cette démarche. Concrètement, l'outil permet :

- aux responsables de créer des groupes de travail,
- aux salariés de proposer des idées d'amélioration en précisant les gains potentiels (économie, ergonomie des postes de travail, sécurité, etc.),
- d'impliquer d'autres personnes dans la réalisation de ces idées,
- aux gestionnaires d'utiliser les exports de l'application pour produire des indicateurs de performance.

Transfert de fichiers

Pour des raisons de sécurité et de maintenance, la transmission de fichiers volumineux (supérieur à dix mégaoctets) ou de certains types de fichiers (exe, bin, zip, etc...) est impossible via la messagerie de l'entreprise. Afin de répondre à ce besoin, une application dédiée a été développée proposant aux salariés de l'entreprise de pouvoir créer un compte en attachant les fichiers volumineux à transférer. Après une analyse systématique des fichiers déposés par un antivirus, ce compte est alors communiqué à la personne située à l'extérieur de l'entreprise, cette personne peut alors à travers une plate-forme Web télécharger et/ou déposer ces fichiers.

Véhicules

Afin de faciliter la gestion des trois parkings de la société par les gardiens, une application a été mise en place offrant la possibilité au gardien :

- d'extraire les données des véhicules déclarés par les salariés,
- de saisir et gérer les véhicules autorisés à stationner sur les différents parkings,
- d'effectuer des recherches croisées pour obtenir les informations en fonction de l'immatriculation du véhicule, du type ou du nom du véhicule ou encore depuis le nom du salarié.

Cette application permet notamment au gardien d'intervenir rapidement si un véhicule non autorisé est stationné sur un parking ou de communiquer efficacement si un problème est survenu sur le véhicule d'un salarié.

ESI

L'application ESI permet au gardien d'obtenir une liste des personnes ESI (Équipier de Seconde Intervention) qui peuvent et doivent intervenir en cas de danger en attendant les sapeurs pompiers. Cette liste est mise à jour en fonction des pointages des salariés (disponible dans le système SAP-HR), afin de proposer au gardien uniquement les personnes présentes dans l'enceinte de la société.

Capacités fonctionnelles

En collaboration avec la médecine du travail et le service des méthodes, un projet a été initié afin d'améliorer l'ergonomie des postes et de réduire la pénibilité des ouvriers de l'usine en affectant ces derniers à des postes adaptés en fonction de leurs capacités physiques et de leurs compétences. Une application informatique a été conçue pour répondre à ce besoin en permettant à trois acteurs d'agir sur l'application : le médecin du travail, le méthodiste et les responsables. Le médecin du travail a pour rôle de déterminer les capacités physiques de la personne en attribuant des notes par rapport à des critères déterminés par le groupe de travail initiateur du projet. De son côté, le méthodiste doit lister, maintenir et définir la pénibilité des postes de travail de la société en attribuant également un ensemble de notes en adéquation également avec les critères exprimés par le groupe de travail. L'application effectue ensuite un rapprochement de ces résultats dans une matrice qui est consultable par les responsables, ces derniers peuvent alors vérifier la liste des postes de travail qui sont adaptés à une personne ou sélectionner les candidats potentiels pour un poste de travail précis.

2.3.3 Retour utilisateur

Depuis fin 2010, un retour utilisateur a été mesuré lors d'une enquête semestrielle sur l'ensemble des utilisateurs de l'entreprise. Lors de cette enquête, deux questions notées de 1 à 6 ont été posées concernant le portail Intranet, la figure 2.10 montre l'évolution de ces résultats au cours de ces dernières années (les notes de 1 à 6 ont été transformées sur un score à 100 points).

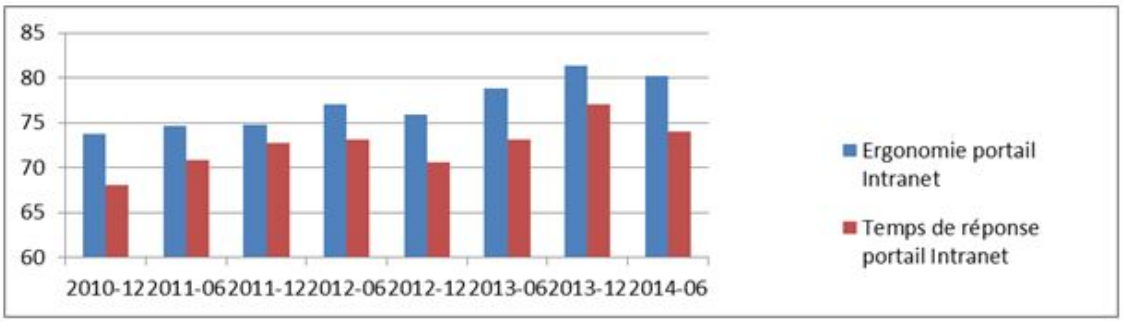


FIGURE 2.10 – Retour utilisateur exprimé sur une échelle de 100 points

On observe une tendance à la hausse de la satisfaction des utilisateurs depuis l'intégration du ZendFramework au sein du portail Intranet. Toutefois cette satisfaction est à relativiser, en effet on évalue uniquement l'ergonomie et la rapidité du site en général. Cette analyse de satisfaction ne tient effectivement pas compte de l'adéquation entre les besoins des utilisateurs et le contenu que propose le site.

2.3.4 Chiffres clés

La figure 2.11 montre l'évolution du nombre de visites sur le portail Intranet ces dernières années. Il est intéressant de noter que le nombre de pages vues est en constante augmentation alors que le nombre de visites a diminué en 2013 et 2014. On peut expliquer ce phénomène par une baisse de l'effectif ces dernières années, ainsi qu'à une augmentation de la quantité de contenu disponible par le portail Intranet.

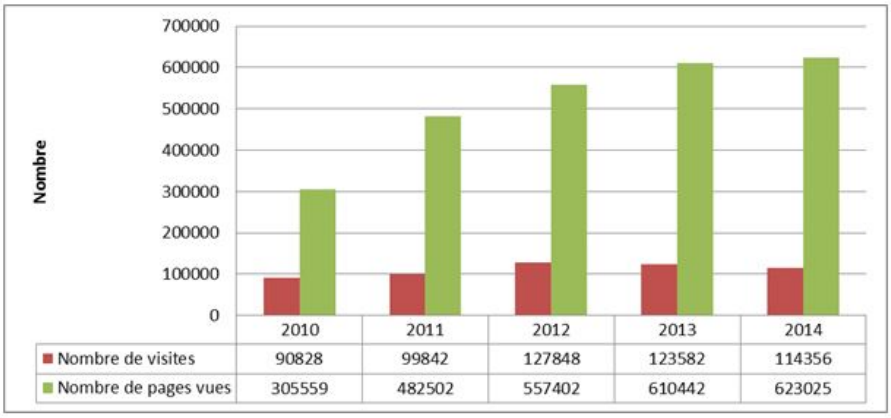


FIGURE 2.11 – Evolution du nombre de visites et des pages vues

Une autre information intéressante qui est illustrée par la figure 2.12 est l'utilisation des applications que l'on peut mesurer par le nombre de pages vues. Au sein de cette figure, les "petites" applications ont été exclues du périmètre d'analyse. On peut noter que la gestion des absences est l'application la plus largement utilisée, ce qui est logique car l'ensemble des salariés ont l'obligation de l'utiliser pour effectuer leurs demandes. Une autre remarque est l'exclusion de l'application gestion des déplacements de cette analyse, en effet comme l'unique utilisateur est la secrétaire de direction, il y a très peu de requêtes sur le serveur, elle est donc considérée comme une "petite" application.

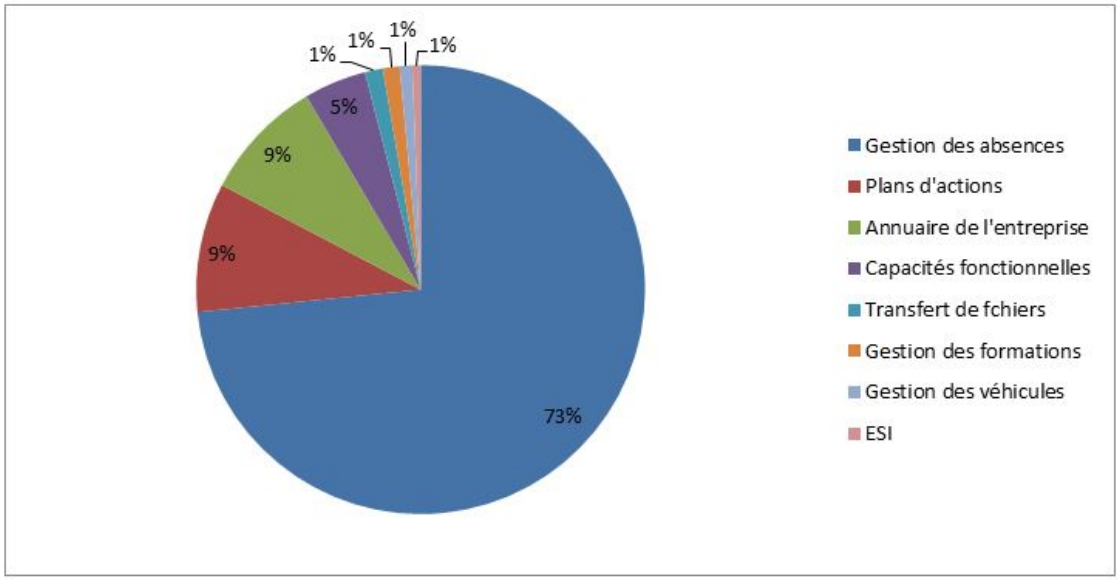


FIGURE 2.12 – Répartition des pages vues par application

3 Cahier des charges

3.1 Rappel du contexte

Le portail Intranet est un PAE développé en 2008 pour la société FERCO et ce dernier n'est plus en adéquation avec les besoins de la société : qualité exigée, productivité, soucis de maintenance et de performances. C'est pourquoi le projet d'élaborer une nouvelle solution de PAE a été décidé.

Le sujet de l'étude est donc de réaliser un nouveau portail applicatif, qui va remplacer le portail Intranet existant, tout en améliorant la qualité des services fournis et en supprimant les limitations du système actuel.

3.2 Analyse de l'existant

3.2.1 Architecture et technologie

Le portail Intranet est une application complexe, modulaire et flexible. Son développement a donc nécessité l'intégration de différentes technologies et librairies pour répondre aux besoins exprimés par nos utilisateurs. Une rapide présentation de ces technologies est réalisée ci-dessous.

3.2.1.1 Architecture MVC

L'Intranet a été développé en respectant une architecture Modèle vue contrôleur (MVC) comme le montre la figure 3.1. Pour rappel, une architecture MVC permet de séparer les composants applicatifs en trois catégories :

— Modèle : Le modèle représente la description, le traitement et la manipulation des données. Il regroupe tous les composants liés à la gestion des données et la base de données se trouve être l'un de ses composants.

— Vue : La vue représente les résultats renvoyés à l'écran de l'utilisateurs. Ces affichages sont généralement issus des données fournies par le modèle. Les vues offrent également la possibilité aux utilisateurs d'interagir avec le modèle en fournissant des composants graphiques pour envoyer des commandes de traitement (bouton, liste de sélection, champs texte, etc.)

— Contrôleur : Le contrôleur a la responsabilité de synchroniser la vue et le modèle. Il réceptionne les événements de la vue (générés via des boutons d'actions par exemple), effectue le traitement logique qui est lié à cet événement et demande au modèle d'effectuer les traitements de mises à jours des données.

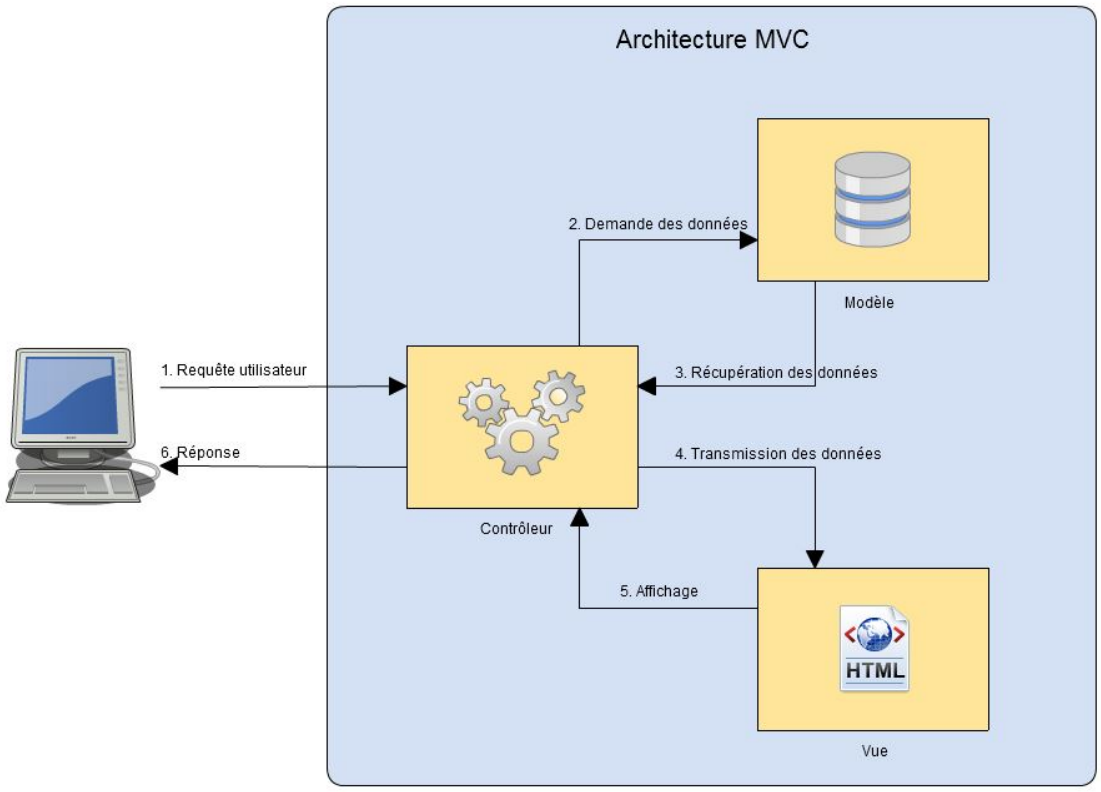


FIGURE 3.1 – Architecture MVC d'une application

Cette séparation des couches à l'aide de l'architecture MVC offre une meilleure lisibilité du code source et permet un regroupement homogène des fonctions de l'application. Ces éléments sont importants et apportent des gains lors du développement et de la maintenance de l'application.

3.2.1.2 Technologies

Le framework

La société Zend Technologies a créée le Zend Framework (aussi appelé ZF ou ZF1) pour offrir à PHP 5 un cadre de travail dans le but de simplifier le développement d'application Web tout en proposant des bonnes pratiques. Ce framework propose nativement d'utiliser l'architecture MVC, mais ne l'oblige pas. De nombreuses solutions PHP existent, toutefois il a été montré que Zend Framework possède des avantages essentiels [PAULI, 2009], voici la liste des principaux avantages :

— une communauté forte qui assure une pérennité de l'outil,
— des concepteurs expérimentés et un code source rigoureusement testé pour une qualité de code garantie, permettant de réduire les risques engagés,
— un support commercial et technique proposé par la société Zend, qui pour rappel est la société dont le domaine d'expertise est l'industrialisation du langage PHP,
— des conventions claires et précises qui vont dans le sens du travail collaboratif, permettant d'augmenter la rapidité de développement et de faciliter la reprise du travail à long terme,
— des composants souples avec peu d'interdépendance,
— un principe de fonctionnement simple qui n'impose pas une structure rigide, il est tout à fait possible d'utiliser librement des composants sans devoir respecter une structure pré-définie.

Le choix de ZF a donc été fait en raison des avantages cités ci-dessus. Néanmoins, ZF nécessite des connaissances avancées en PHP et de bonnes notions sur le développement logiciel afin de l'implémenter efficacement. Les points clés de la solution du ZF sont illustrés à l'aide de la figure 3.2.

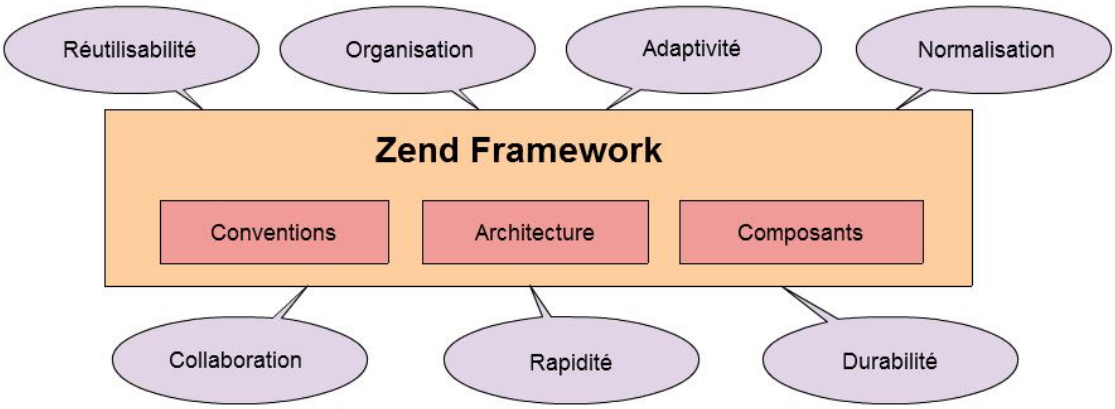


FIGURE 3.2 – Zend Framework

Doctrine 1

Doctrine est un object-relational mapping (ORM) pour le langage PHP. L'ORM est une technique qui consiste à donner la possibilité aux développeurs de travailler « virtuellement » avec une base de données orientée objet à partir d'une base de données relationnelle. Ceci est possible en définissant, et en configurant des correspondances entre la base de données et les objets de l'application. Finalement, les développeurs ne vont plus utiliser le langage SQL directement, ils manipuleront à la place, les objets qui vont reproduire les modifications dans la base de données. Doctrine permet de faire une abstraction de la base de données, tout en travaillant directement avec des objets (appelés entités) en PHP.

jQuery

JQuery est une librairie développée en JavaScript dont le principal objectif est de gagner du temps dans le développement des applications, il s'agit même de son slogan [jQuery] : "write less, do more", effectivement la mise en place de jQuery permet de réduire considérablement le code. Voici un aperçu des possibilités qu'offre cette librairie JavaScript :

— parcourir et modifier le DOM,
— gérer les événements de la page Web,
— gérer les effets et les animations de la page Web,
— manipuler les feuilles de style en cascade,
— gérer les requêtes AJAX.

Une autre force de jQuery est qu'il est cross-browser, c'est à dire qu'il est compatible avec tous les principaux navigateurs du marché (Internet Explorer, Firefox, Chrome et Opéra), ceci est un énorme avantage car un code unique fonctionnera quelque soit l'environnement sans avoir de recours à des adaptations spécifiques pour un navigateur précis. Une seule fonction nommée "$" permet d'accéder à l'ensemble des API, il est également possible de réaliser un "chainage d'actions", c'est à dire d'écrire

en une seule ligne ce que l'on ferait avec dix en passant par des étapes intermédiaires, ce point permet de simplifier le développement en réduisant considérablement le nombre de lignes du code source.

XHTML

XHTML (Extensible HyperText Markup Language) est le langage de structuration de l'information qui a été utilisé pour écrire les pages Web de l'application. Il se différencie du langage HTML en se basant sur la syntaxe définie par XML plutôt que le SGML.

3.2.1.3 Structure des fichiers

Le portail Intranet a été développé avec le framework ZF1, ce dernier propose une architecture modulaire avec pour objectif de développer des applications Web. Le portail Intranet a donc respecté ce principe d'architecture, avec une structure de fichiers qui permet de séparer les applications développées en "module". La figure 3.3 illustre la structure des fichiers utilisée par le portail Intranet.

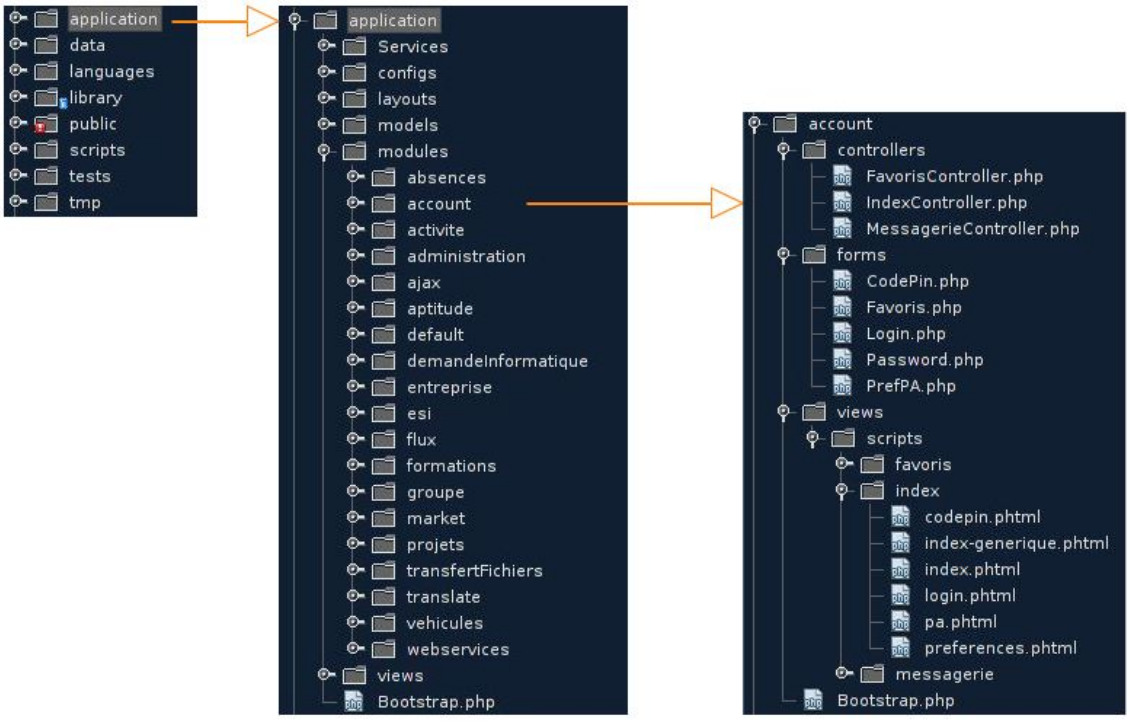


FIGURE 3.3 – Structure des fichiers du portail Intranet

application Le dossier application est le plus important, en effet ce dernier possède les éléments suivants qui sont indispensables au bon fonctionnement de l'application :

- services : partie de l'application qui contient les services de validation des entités (validation de format et de logique métier).
- configs : ce dossier stocke les différents fichiers de configuration de l'application sous un format de fichier .ini.
- layouts : contient le "layout" de l'application, c'est à dire le modèle de présentation qui va être enrichi par les vues afin de délivrer la page de présentation complète.
- models : contient les "entités" gérées par l'ORM Doctrine 1.

— modules : le dossier modules contient l'ensemble des applications que propose le portail Intranet. Chaque dossier module est composé de "controllers" qui vont définir la logique métier, de "views" qui vont servir à définir l'affichage des pages de présentation et de "forms" qui définissent les formulaires de saisies permettant la saisie d'informations par les utilisateurs.
— views : ce répertoire contient les pages de présentation de l'application, ces dernières peuvent être partagées entre différents modules.
— Le fichier Bootstrap.php : littéralement le fichier d'amorçage de l'application. Ce dernier contient l'ensemble des directives qui permettent d'initialiser l'intégralité des composants qu'utilise l'application : gestion des logs, accès, mails, vues, initialisation de la base de données. Ce fichier est la noyau de l'application.

data Dans ce dossier est enregistré les différents éléments temporaires ou définitif de l'application. On retrouve notamment des éléments comme les données de sessions utilisateurs, des pièces jointes et des logs.

languages La société FERCO faisant partie d'un groupe international, le portail Intranet a été prévu pour pouvoir être internationalisé c'est à dire avoir la capacité à afficher ses pages dans différentes langues. Les traductions de l'application sont disponibles dans ce dossier.

library L'ensemble des frameworks et librairies sont contenus dans ce dossier.

public Toutes les ressources disponibles directement par le navigateur sont stockées dans le dossier public. On y retrouve les fichiers images, javascripts et css de l'application. Afin de rendre accessible ces ressources et de protéger le reste de l'application, des directives .htaccess configurent la sécurité de ces dossiers.

scripts Le dossier scripts contient l'ensemble des jobs de l'application. Ces jobs permettent généralement la mise à jour de données applicatives ou prennent en charge la notification par mail de certains événements (rappel de validation, traitement ou encore envoi de rapports hebdomadaires).

tests Ce dossier stocke les résultats de tests.

tmp Les éléments temporaires sont enregistrés dans ce dossier.

3.2.2 Interfaces avec le système d'information

Le portail Intranet est interfacé avec plusieurs éléments du système d'information. La figure 3.4 présente les interfaces du portail Intranet.

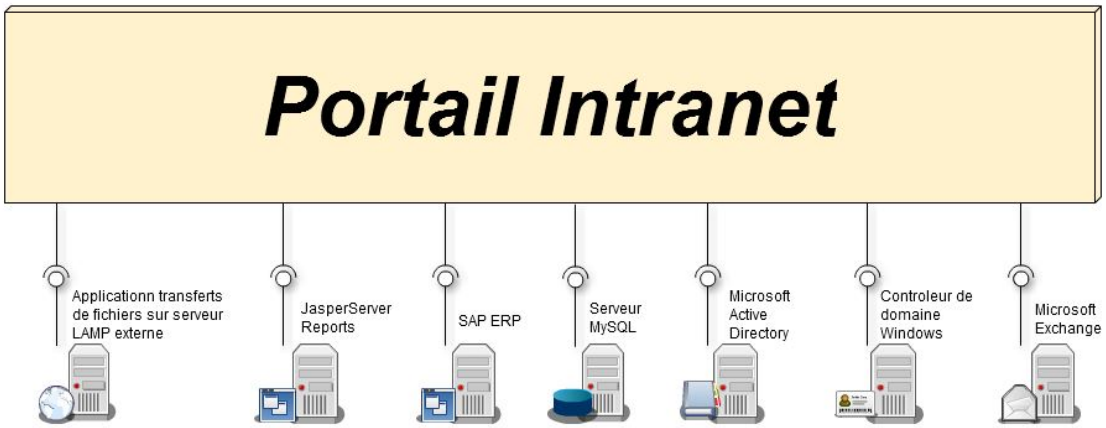


FIGURE 3.4 – Interfaces du portail Intranet

Application transferts de fichiers

Une suite d'API est disponible via le protocole HTTP (en utilisant la technique d'architecture REST) depuis le serveur Web externe fercubu03. Ces API permettent de gérer la transmission des fichiers déposés par les salariés de FERCO à leurs contacts situés à l'extérieur de l'entreprise. Le portail Intranet utilise la librarie Zend_Http_Client pour communiquer avec ce serveur externe.

JasperServer Reports

JasperServer Reports est le logiciel de Business Intelligence (BI) permettant la génération de reporting depuis différentes sources de données (fichiers Excel, base de données, fichiers CSV, etc.). Ce logiciel est interfacé avec le portail Intranet à travers un ensemble de webservices simple object access protocol (SOAP). Le portail Intranet utilise ces webservices pour demander le résultat de l'exécution de certains rapports.

SAP ERP

Le progiciel SAP contient une grande partie des données du système d'informations de l'entreprise. Le portail Intranet peut accéder à ces données à travers l'appel de fonctions RFC. Le portail se connecte à deux destinations : le serveur qui gère les données des ressources humaines (SAP-HR) et le serveur de production qui permet la gestion des données de production (SAP-AG1).

Serveur MySQL

Le serveur MySQL est le système de gestion de base données (SGBD) utilisé par le portail Intranet pour stocker les données applicatives. Le portail Intranet utilise l'ORM Doctrine 1 pour manipuler la base de données en donnant l'illusion de travailler avec une base de données orientée objet.

Microsoft Exchange

Microsoft Exchange est le serveur de messagerie utilisé au sein de l'entreprise FERCO. L'envoi des mails est possible depuis le portail Intranet en utilisant le protocole SMTP à travers la libraire Zend_Mail du framework PHP Zend Framework 1.

3.2.3 Contraintes de réalisation

Le portail Intranet est une application qui offre différents services sous forme d'applications Web. Il gère donc l'ensemble des problématiques liés à ce type de développement : contrôle des accès, gestion des formulaires de saisie, interface graphique, etc...

L'Intranet a été développé avec un ensemble de contraintes qu'il est nécessaire de prendre en compte dans le but d'élaborer une nouvelle solution.

Connexion automatique à l'application

Les ordinateurs de l'entreprise font partie d'un domaine Windows et les utilisateurs se connectent à l'aide d'un compte informatique, qui est authentifié par un Active Directory lors de l'ouverture de leurs sessions. Dans le but de simplifier la connexion au portail Intranet, il a été mis en place un système de connexion automatique : Single sign on (SSO). Ce système implique que la personne n'a pas besoin de ressaisir ses identifiants car ces derniers ont déjà été précédemment vérifiés par l'Active Directory, nous connaissons donc la personne connectée sur le poste de travail.
Afin de pouvoir effectuer une connexion automatique avec le domaine Windows, l'application utilise le protocole NT Lan Manager (NTLM).

Connexion par badge ou mot de passe

Une partie de la population de l'entreprise sont des ouvriers, ces derniers ne disposent pas de compte nominatif, c'est pourquoi ils utilisent une "borne" (ordinateur en libre-service) pour plusieurs personnes. Cette organisation implique qu'il est nécessaire d'avoir des méthodes de connexion supplémentaires pour permettre l'authentification de ces personnes.

Connexion par badge
Le système de sécurité par badge utilise un pistolet code barre afin de lire le code barre imprimé sur le badge, Le texte lu par le système est envoyé à un formulaire, puis l'analyse du temps de saisi dans le formulaire constitue le contrôle. La connexion par badge se trouve être la moins sécurisée pour plusieurs raisons :

- Les ouvriers ont tendance à laisser leurs badges sur le poste de travail (car ils déclarent la productivité sur leur poste à l'aide du badge), une autre personne peut donc subtiliser le badge et usurper l'identité de l'ouvrier.
- Si le salarié connaît le type de code barre utilisé, ainsi que le texte se trouvant derrière le code barre, il peut alors reproduire un code barre similaire et donc usurper la personne.
- L'analyse du temps de frappe est effectuée en parti par du Javascript, ce code qui est disponible côté client peut donc être altéré facilement.

Pour toutes ces raisons de sécurité, la connexion par badge ne donne pas accès aux données sensibles des applications.

Connexion par mot de passe
La connexion par mot de passe est délivrée à certains ouvriers qui sont autorisés à accéder à des contenus sensibles de certaines applications (données médicales, suivi de salariés, etc...).

Afficher les informations d'un utilisateur en fonction du contexte applicatif

Lors d'un clic sur le nom d'un salarié, une fiche descriptive de la personne est affichée à l'écran via une popup géré via la librairie jQuery. Les informations affichées dans ce cadre sont différentes en

fonctions de l'application où l'utilisateur se trouve. Ci-dessous quelques exemples qui illustrent cette fonctionnalité :

- Gestion des absences : On affiche les soldes de la personne, ainsi que les absences passées, en cours et futures.
- Véhicules : Les véhicules du salariés sont affichés.
- Capacités fonctionnelles : La carte des capacités fonctionnelles est affichée (contraintes physiques de la personne, ainsi que ses capacités physiques).

Export Excel

Les utilisateurs avancés utilisent le logiciel Microsoft Excel pour effectuer des analyses et rapports supplémentaires. Le portail Intranet prévoit l'export de ses données applicatives sous ce format, afin de donner aux utilisateurs un accès simple à leurs données, ainsi que de la transparence et de la visibilité.

Export PDF

Certaines applications génèrent différents documents (rapports, convocations et modèles), l'application permet donc l'export et la mise en forme des données sous le format PDF.

Envoi d'alertes par mail

Des alertes de validation sont envoyées par mail aux différents responsables pour les rappeler de valider les demandes qui restent en attente dans le système. Ces rappels sont envoyés par des jobs de traitements spécifiques via des tâches planifiées du serveur.

Jobs de traitement

les jobs de traitement sont des scripts qui sont planifiés et exécutés par le système et qui permettent de réaliser une tâche spécifique : envoi de notifications, de rappels, modification des données, etc...

Téléchargements de fichiers

Certaines applications permettent l'ajout et le retrait de pièces jointes, il faut aussi donner la possibilité aux utilisateurs de gérer simplement des fichiers.

Système de reporting

Outre les exports Excel qui permettent l'accès des données de base aux utilisateurs, d'autres rapports sont disponibles dans les applications. Des outils différents peuvent être utilisés pour générer ces rapports :

- Le système de JasperServer Reports : ce système permet de réaliser des graphiques plus avancés, mais nécessite plus de complexité à être mis en place.
- Un export Excel : si l'export demandé est simplement un cumul, une moyenne ou encore une agrégation quelconque des données, alors un export applicatif (plus simple à réaliser qu'un rapport Jasper) paraît plus pertinent.
- un export PDF : pour certains rapports simples qui vont être imprimés et affichés, on peut utiliser une extraction des données et générer un fichier PDF. Ceci est par exemple utilisé dans le cadre de la gestion des plans d'actions, les responsables impriment les actions en cours de traitement de leurs collaborateurs pour effectuer le suivi de leurs réunions.

Système de workflow

Plusieurs applications utilisent un système de workflow : les demandes sont créées puis validées par différents acteurs afin de suivre et respecter un processus de traitement. Actuellement aucun moteur de workflow n'est utilisé pour gérer cet aspect, le workflow est donc entièrement géré par le code source des contrôleurs de l'architecture MVC.

3.2.4 Les limitations

Le portail Intranet a été une réponse adaptée lors de son développement aux besoins des utilisateurs, toutefois ce portail montre des limitations sur plusieurs aspects qui vont être présentés ci-dessous.

L'efficacité des développements

Les besoins des utilisateurs continuent à évoluer et sont plus nombreux au fil du temps. Les technologies utilisées par le portail Intranet ne suffisent plus à répondre de manière performante aux problématiques posées (des workflow plus complexes, des formulaires plus lourd à gérer, une interface graphique avec des limitations, etc.). En effet, on peut citer plusieurs éléments du portail Intranet qui devraient être modifiés pour atteindre une meilleure productivité des développements :

- La gestion des messages de logs d'erreurs : ils sont gérées par le développeur via un composant du framework ZF1, le Zend_Log, sa gestion est intégralement définie par le développeur (mise en place dans un fichier, dans une base de données, alerte par mail, filtre par niveau de criticité, etc.). Le framework ne délivre pas de système de messages de logs standard propre au framework, cette limitation est gênante pour le développeur qui ne dispose pas d'outils efficaces pour analyser les événements propres au framework. Il est donc plus difficile lors du développement de trouver certaines erreurs qui sont générées par le framework.
- L'absence d'outil qui aide à la traçabilité de l'application. Le seul moyen de tracer une erreur est la pile d'erreurs remontée par une exception.
- Le manque de lien entre les différents composants de l'architecture MVC. Il serait effectivement par exemple plus efficace de pouvoir lier un formulaire HTML à une entité du modèle et ainsi avoir la possibilité d'alimenter simplement un objet du modèle.
- La génération des entités est un autre aspect qui pourrait être amélioré par l'utilisation d'un outil de génération, actuellement pour créer une nouvelle entité, il faut écrire manuellement la classe, ainsi que l'ensemble des annotations nécessaires.

Cette liste n'est pas exhaustive et montre avant tout des limitations du framework MVC utilisé, c'est à dire ZF1. La qualité du framework n'est pas réellement ce qui est remis en cause ici, mais plutôt sa philosophie d'origine qui est de ne pas livrer un socle cohérent pour créer une application Web, en effet sa philosophie était de fournir un ensemble de librairies indépendantes permettant la création et la gestion d'une application Web. Ce point est important, effectivement ce framework laisse une grande liberté aux développeurs qui peuvent choisir quels composants utilisés ou non pour tel besoin, mais en contrepartie le développeur souffre d'un manque de solutions "clés en main" pour régler ses problèmes. De plus ce framework ne possède pas d'outils efficaces pour orchestrer, superviser et utiliser les différents composants du framework, ce qui est au quotidien pénalisant pour le développeur.

La maintenance

L'organisation et la structure des fichiers du portail Intranet a l'avantage de définir une arborescence, cependant sur certains aspects elle entraîne une mauvaise gestion du code source, ce qui amène une maintenance plus compliquée. On peut citer par exemple, un formulaire (défini par une classe PHP), où la partie Javascript liée à ce formulaire est stockée dans cette même classe PHP, nous avons donc deux technologies différentes au sein du même fichier (même si ces parties de code traitent le même aspect ce n'est pas une bonne pratique).

Un autre point qui complique la maintenance est la diversité des développeurs qui sont intervenus sur ce PAE avec des niveaux de programmation différents, la qualité du code source est donc de niveau hétérogène sur l'ensemble du PAE. Il est aussi à noter que l'absence de règles de développement clairement définies a amplifié ce phénomène.

Les performances du système

Au départ du portail Intranet, une page était affichée en environ une seconde, suite à l'augmentation du nombre de framework et de librairies utilisés, ainsi qu'à l'augmentation des informations et fonctionnalités disponibles sur les pages Web, le temps de génération a augmenté jusqu'à atteindre les trois secondes pour le même type de page. De manière générale, les framework permettent d'augmenter les possibilités et la qualité du site en général, cependant il y a eu un impact sur les performances (point qui a été remonté par les utilisateurs dans les enquêtes de satisfaction).

Toujours dans les soucis de performances, un autre élément qui intervient dans ces problèmes de lenteur est le système de routage de ZF1, ce dernier n'utilise pas un système de routage par configuration pour la gestion de l'aiguillage de la requête, en effet le système analyse l'URL et va déterminer automatiquement le contrôleur et l'action de l'architecture MVC qui sont à appeler. Pour effectuer cette détermination, ZF1 va effectuer une inspection des fichiers et analyser les classes qui y sont contenus pour déterminer le contrôleur et l'action à invoquer. Cette méthode permet au développeur de s'affranchir de la configuration d'un système de routage, mais cette "magie" à un prix : la performance est réduite, la lisibilité du système de routage est faible et cela impose une convention de nommage et une arborescence très stricte pour que le routage soit fonctionnel.

La gestion de la mobilité

Le portail Intranet n'intègre aucune gestion de la mobilité, l'interface graphique est donc identique si l'on se connecte via un smartphone, une tablette ou un ordinateur. Comme nos utilisateurs possèdent ce genre d'équipements, il serait intéressant de pouvoir proposer un rendu adapté à l'ensemble de leurs périphériques.

Le système de workflow

Aucun système de workflow n'est présent dans le portail Intranet, l'ensemble de la gestion du workflow est donc réalisé par les contrôleurs de l'architecture MVC. Ceci favorise un mauvais découplage des fonctions, ainsi que des soucis de lisibilité du code source des contrôleurs et entraîne par conséquent des problèmes de maintenance.

3.3 Étude d'une nouvelle solution

3.3.1 Spécifications fonctionnelles

La nouvelle solution à élaborer doit garder la même interface avec le système d'information et les mêmes contraintes techniques (voir parties 3.2.2 et 3.2.3).

Les points suivants sont ajoutés aux besoins exprimés précédemment dans la partie 3.2.3 :
— gérer le problème de la mobilité : c'est à dire pouvoir se connecter avec des périphériques de tailles différentes (smartphone, tablette, PC),
— donner la possibilité à l'administrateur du portail de gérer les droits des personnes sur les applications et les objets métiers à l'aide d'une interface prévue à cet effet,
— informer les utilisateurs des changements survenus sur le portail applicatif via un message propre au sein du portail avec accusé de lecture,
— garder l'historique de toutes les modifications sur les objets métiers gérés par les applications, ceci afin de garder une traçabilité des actions,
— permettre à l'administrateur de pouvoir se connecter à la place d'un utilisateur pour analyser et débloquer des situations,
— améliorer les performances générales par rapport à l'ancien système, avec comme objectif un temps de génération de page inférieur à une seconde (situé entre deux et trois secondes sur l'ancien système) pour des pages "simples", c'est à dire afficher une page avec la lecture de quelques entités de la base de données.

L'élaboration d'un nouveau PAE implique la construction d'une base technique solide, souple et flexible pouvant répondre aux nouveaux développements demandés par les utilisateurs. Dans le cadre du projet, il est nécessaire de déterminer une application pilote à développer et à intégrer au PAE dans le but de pouvoir utiliser et tester efficacement l'ensemble des fonctionnalités de ce dernier.

3.3.1.1 Description fonctionnelle de l'application pilote : la gestion des déplacements

Le sujet qui a été retenue comme application pilote est la gestion des déplacements de la société. Ce sujet est intéressant à traiter car la version disponible dans le portail Intranet ne répondait plus aux attentes du nouveau gestionnaire : pas de reporting adapté, des données manquantes et aucun workflow n'est en place. Les demandeurs continuent d'utiliser le formulaire papier afin de soumettre leurs demandes et une saisie est ensuite réalisée dans le système par le gestionnaire. La réalisation de cette application dans le nouveau PAE permettra de dématérialiser totalement ce support papier en proposant la saisie des informations directement au demandeur.

La création de cette application au sein du PAE va permettre d'utiliser un périmètre complet du socle technique, en abordant les points suivants (non exhaustifs) :
— la gestion des droits et des permissions sur les pages et les objets de gestion,
— les extractions et états (reporting Excel et PDF) à réaliser pour le gestionnaire,
— les formulaires de saisie complexes à réaliser,
— des jobs de traitements pour l'intégration des données dans le système externe de la gestion des temps,
— un système de workflow à mettre en place,
— des notifications à envoyer par mail aux utilisateurs pour les informer de l'avancement de la demande.

Acteurs

Les acteurs concernés par le processus de la gestion des déplacements sont les suivants :

Gestionnaire Le gestionnaire aura la tâche d'effectuer le suivi des déplacements et de procéder à différentes actions, dont notamment :

— réserver les hôtels,
— effectuer la gestion des moyens de transports (acheter des billets, réserver des voitures de société, etc.),
— ajouter les documents de voyage.

Demandeur Il s'agit de la personne qui demande le déplacement, il peut renseigner les informations du déplacement et indiquer quels vont être les autres participants. Cet acteur peut également créer un déplacement pour d'autres personnes sans y participer.

Responsable du demandeur (ResponsableD) Le responsable du demandeur est la personne qui valide le déplacement du demandeur. Il a accès en validation aux demandes créées par ses collaborateurs.

Supérieur du demandeur (SuperieurD) Le supérieur du demandeur est un responsable qui se trouve dans la hiérarchie du demandeur sans toutefois être son responsable direct (responsable du responsable). Il possède les mêmes droits que le responsable, toutefois comme il ne s'agit pas de l'approbateur direct aucune notification n'est envoyée lors de la création d'un déplacement.

Participant Le participant est une personne qui peut uniquement consulter les demandes de déplacement dont il fait parti. Ce dernier est aussi notifié lors de la création ou d'une mise à jour d'un déplacement où il est concerné.

Responsable du participant (ResponsableP) Le responsable d'un participant est notifié lorsque l'un de ses collaborateurs participe à un déplacement. Il peut également consulter les demandes où ses collaborateurs participent.

Supérieur du participant (SuperieurP) Le responsable d'un participant peut accéder en consultation aux demandes où ses collaborateurs sont concernés.

La liste des acteurs peut être représentée graphiquement avec un diagramme d'utilisation (voir la figure 3.5)

Diagramme des cas d'utilisations

Pour des soucis de lisibilité, le cas d'utilisation "Gérer demande" de la figure 3.5 sera détaillé plus loin sous forme de scénarios.

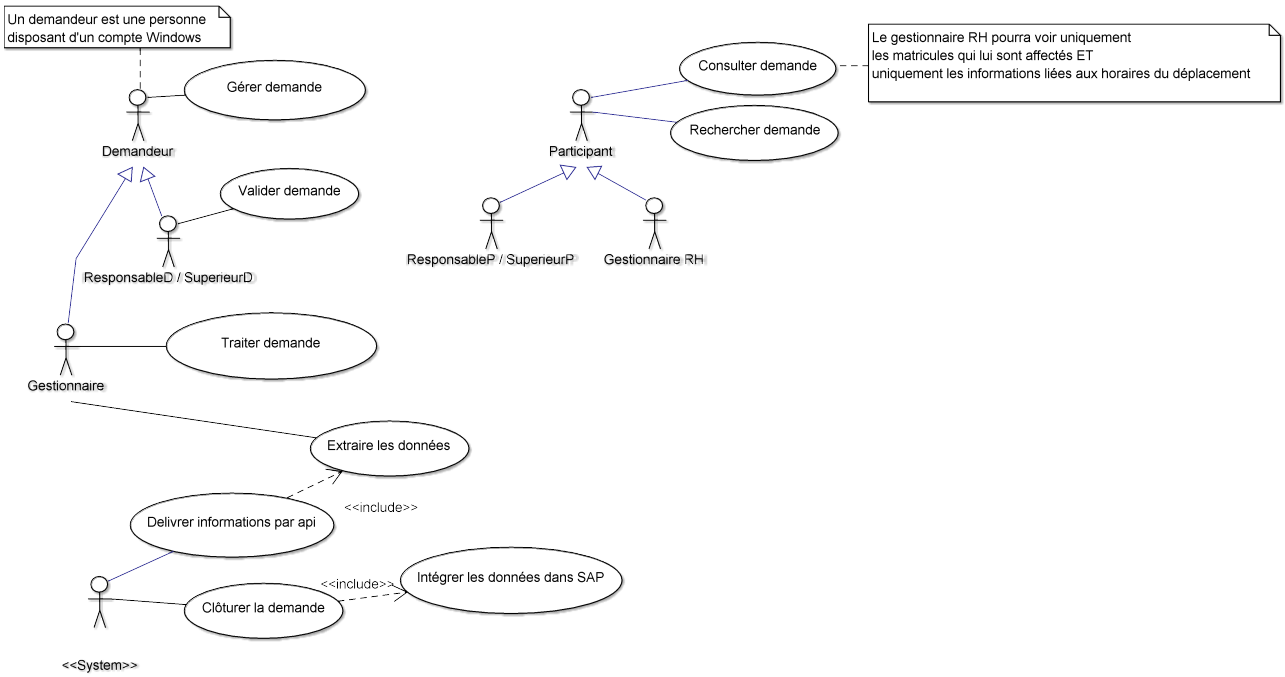


FIGURE 3.5 – Cas d'utilisations de la gestion des déplacements

Les scénarios

Comme mentionné précédemment, le cas d'utilisation "Gérer demande" peut être précisé avec les scénarios suivants :

— Consulter demande,
— Créer demande,
— Modifier demande,
— Supprimer demande.

Consulter demande Le scénario "Consulter demande" est illustré par le diagramme de séquence de la figure 3.6.

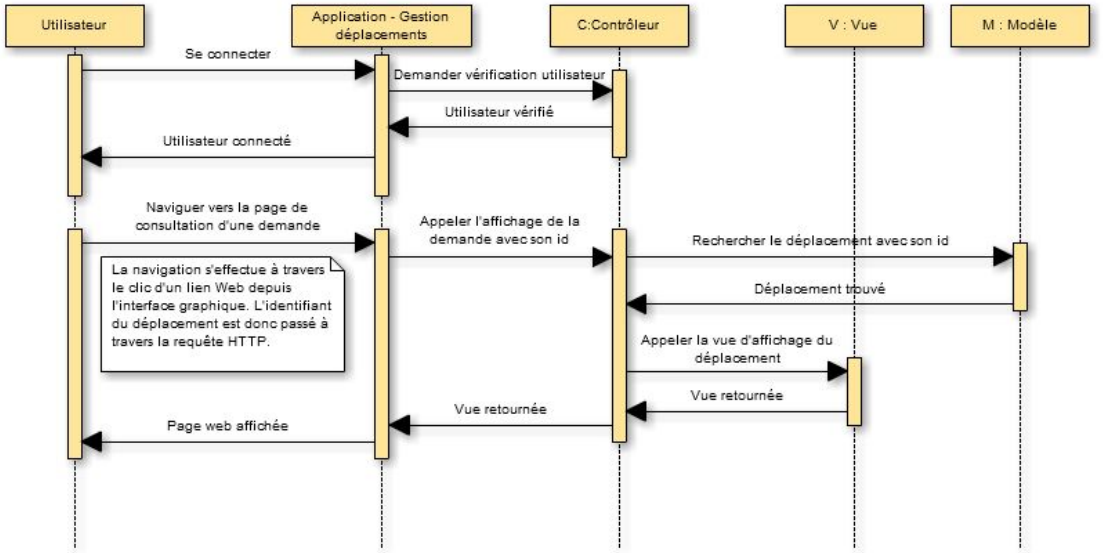


FIGURE 3.6 – Scénario de la consultation d'une demande

Créer demande Le scénario "Créer demande" est illustré par le diagramme de séquence de la figure 3.7.

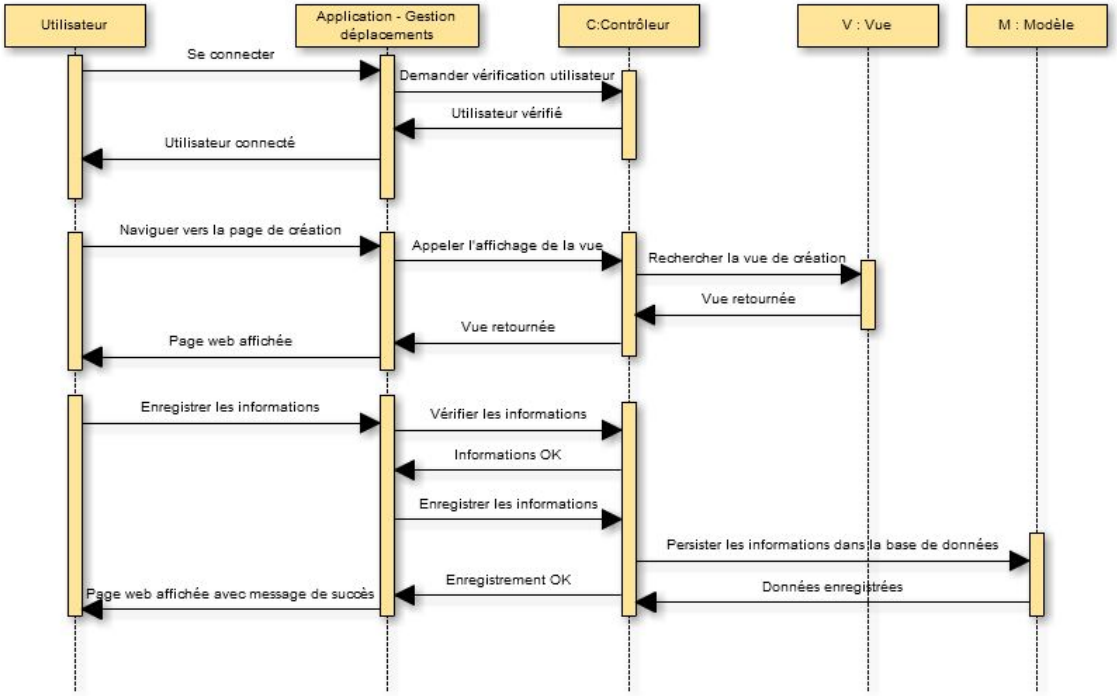


FIGURE 3.7 – Scénario de la création d'une demande

Modifier demande Le scénario "Modifier demande" est illustré par le diagramme de séquence de la figure 3.8.

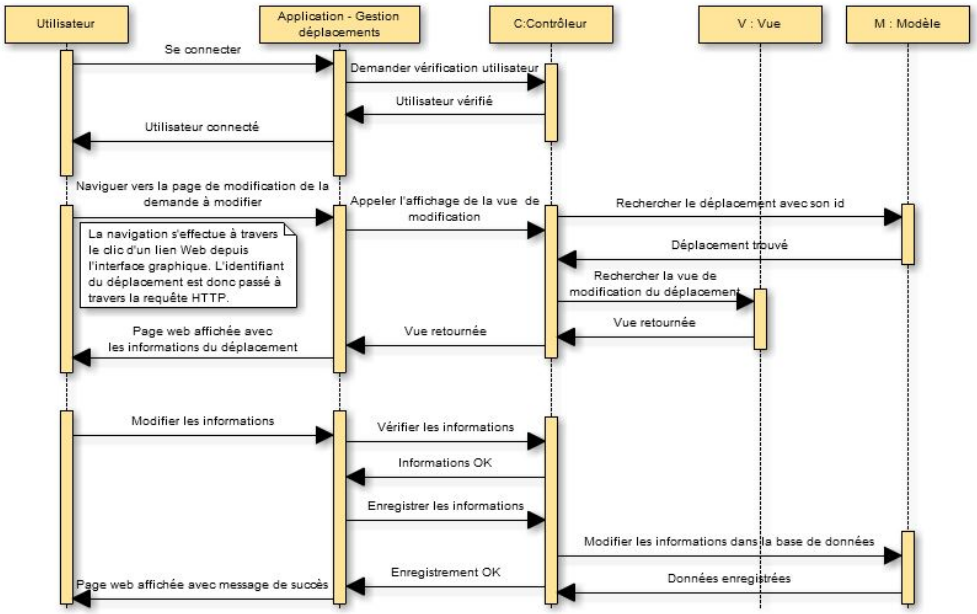


FIGURE 3.8 – Scénario de la modification d'une demande

Supprimer demande Le scénario "Supprimer demande" est illustré par le diagramme de séquence de la figure 3.9.

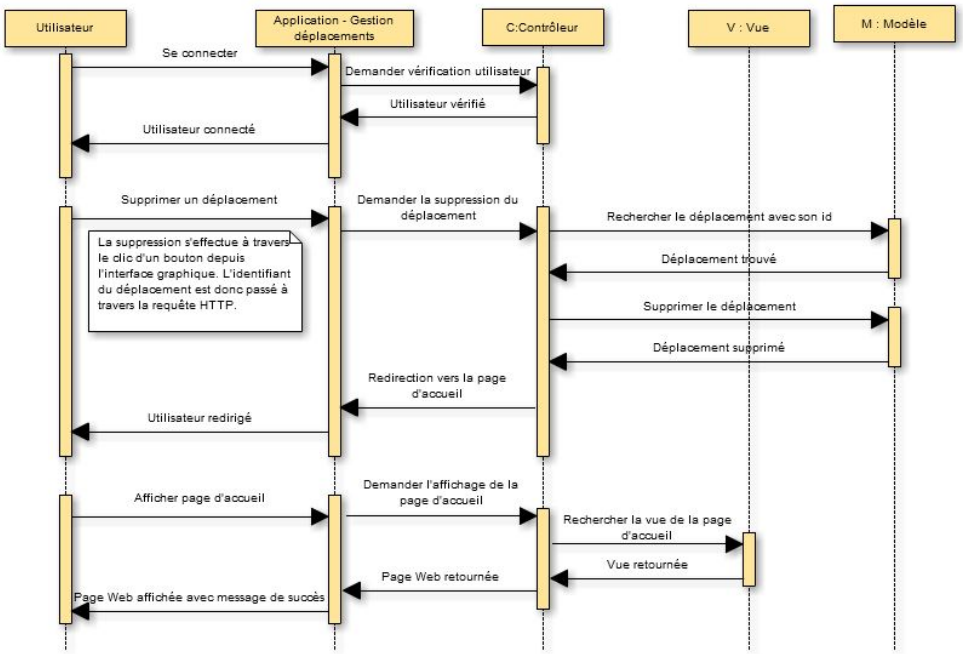


FIGURE 3.9 – Scénario de la suppression d'une demande

Processus de la gestion des déplacements

Le diagramme d'activité de la figure 3.10 permet d'avoir une vue simplifiée du processus qui est traité par l'application pilote.

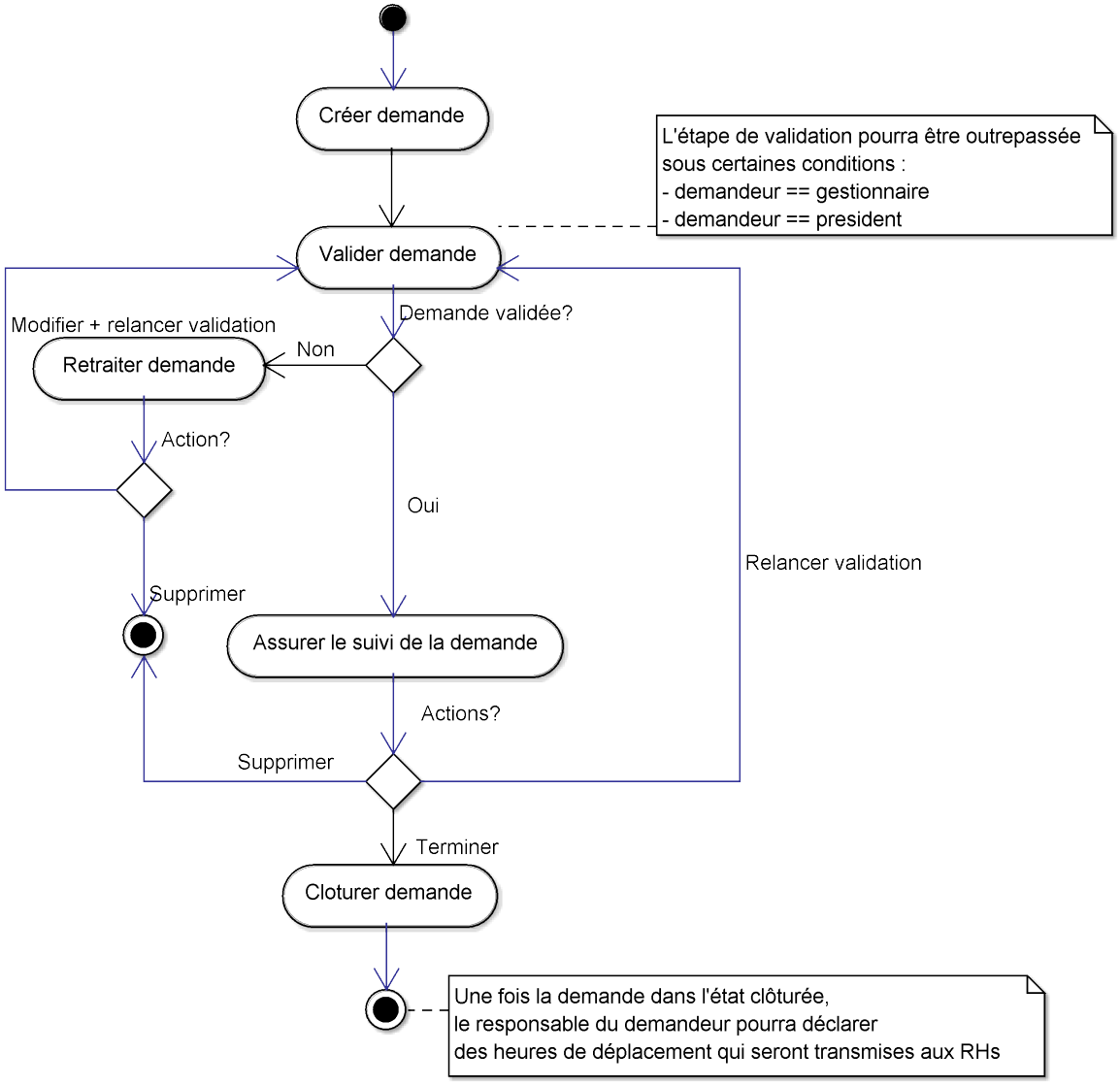


FIGURE 3.10 – Processus simplifié de la gestion des déplacements

On peut remarquer que pour certaines étapes du diagramme d'activité de la figure 3.10, une notification doit être envoyée :

— Créer demande : un mail est envoyé au responsable lui indiquant qu'une nouvelle demande est présente dans le système et nécessite sa validation.

— Valider demande : lors de la validation de la demande par le responsable, un mail d'information peut être envoyé au demandeur en cas de refus (ce dernier a la possibilité de relancer la demande après l'avoir modifiée), dans le cas contraire un mail est envoyé aux participants et

à leurs responsables indiquant que le déplacement est validé.
— Relancer validation : si le demandeur modifie sa demande et souhaite la soumettre une nouvelle fois dans le système, le responsable reçoit un mail demandant son approbation.
— Demande terminée : lorsque le gestionnaire a réalisé toutes les tâches liées au déplacement, un mail est ensuite transmis aux participants et au demandeur indiquant que le déplacement est terminé et qu'ils peuvent maintenant accéder à leurs documents de voyage.

Règles de gestion

Des règles de gestion ont été exprimées par les demandeurs du projet dans le but de fournir un cadre cohérent à l'application :

1. Au minimum un participant par demande de déplacement.
2. La demande de déplacement est sujet à un type de déplacement (visite client, visite fournisseur, séminaire, etc.) et son organisation peut donner lieue à la réservation de plusieurs hôtels, moyens de transports (train, avion, etc.) et éventuellement encore des voitures de location.
3. Chaque élément de réservation génère une tâche pour le gestionnaire qui va devoir réaliser la réservation, une demande de déplacement peut être déclarée comme terminée uniquement si toutes ses tâches ont été traitées par le gestionnaire.
4. La demande de déplacement est créée par un salarié de FERCO et doit être validée par un responsable de sa hiérarchie. Cependant quelques exceptions à cette règle générale existent :
   — Si un avion est présent dans un moyen de transport, la validation doit être effectuée par un cadre dirigeant (dans le but de contrôler les coûts).
   — Si un des participants de la demande est un cadre dirigeant, alors la demande doit être validée par un cadre dirigeant.
   — Si le demandeur n'a pas de responsable (cas du président de la société) alors la demande est automatiquement validée.
   — Si le demandeur est le gestionnaire de l'application, alors la demande est également automatiquement validée.
5. Le gestionnaire peut relancer une demande de validation si le périmètre de la demande de déplacement est modifié (changement du type de déplacement par exemple).
6. Le gestionnaire peut modifier la demande uniquement si elle se trouve dans un état autre que "clôturée" ou "terminée". Cependant le gestionnaire peut repasser une demande de l'état "terminée" à "validée" afin de pouvoir effectuer des modifications (ce cas rare arrive uniquement si le périmètre du déplacement à la dernière minute).
7. L'intégration dans le système SAP-HR des informations de temps ne portera que sur les demandes qui se trouvent dans un état "terminée" et dont le gestionnaire ou le système demande la clôture de la demande. Suite à cette action de clôture le système imposera l'état "clôturée" à la demande.
8. La personne qui a validée la demande peut demander à modifier le nombre d'heures qui a été comptabilisé dans le système SAP-HR (cette action est possible uniquement pour les demandes qui sont dans un état "clôturée"). En effet lorsque la demande est clôturée automatiquement par le système SAP-HR, ce dernier insère automatiquement les heures du déplacement en fonction des horaires théoriques de la personne, hors pour certains cas où la personne va dépasser ses heures théoriques de présence durant un déplacement, on doit pouvoir demander la modification des heures qui ont été imputées. Cette déclaration est possible jusqu'à sept jours après la date de clôture du déplacement par le système.

Diagramme d'états-transitions

Le cycle de vie d'une demande de déplacement peut être décrit par un diagramme d'états-transitions, la figure 3.11 permet de suivre ces différents états.

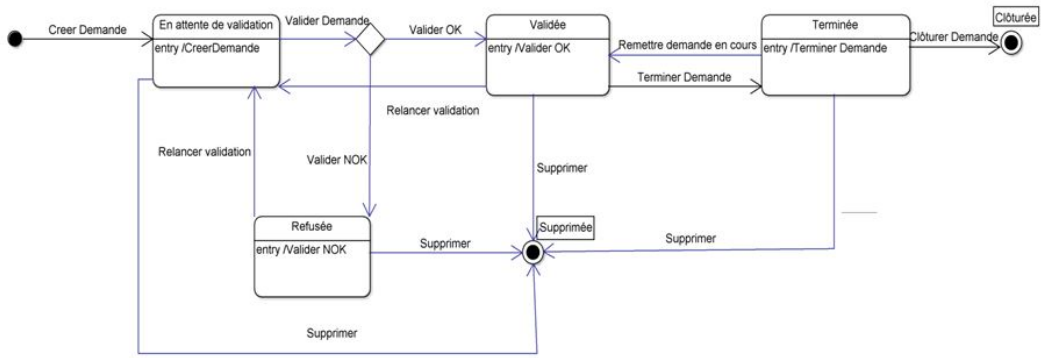


FIGURE 3.11 – Diagramme d'états-transitions d'une demande de déplacement

Comme le montre la figure 3.11, la demande est initiée suite à un événement de création et se trouve dans un état "En attente de validation".

Pendant cette étape de validation, un approbateur peut accepter ou refuser la demande. Si une demande est refusée par un approbateur, elle se trouvera dans un état "Refusée". Le demandeur pourra alors selon son choix, soit modifier la demande et ainsi la remettre dans l'état "En attente de validation", soit simplement la supprimer du système.

Dans le cas contraire où la demande a été acceptée lors de la phase de validation, cette dernière se trouvera dans l'état "Validée" et pourra être traitée par le gestionnaire de l'application. Une fois que ce dernier aura terminé son traitement, la demande passera dans un état "Terminée" et pourra être clôturée par le système qui imposera l'état "Clôturée" à la demande.

Maquettes

Les maquettes suivantes permettent d'avoir une vue du résultat à obtenir pour l'utilisateur. D'un côté plus technique ces maquettes offrent un première synthèse des données à intégrer dans le Modèle conceptuel de données (MCD) de l'application.

Saisie des informations générales du déplacement

La figure 3.12 illustre une maquette possible pour la création d'un déplacement dans le système.

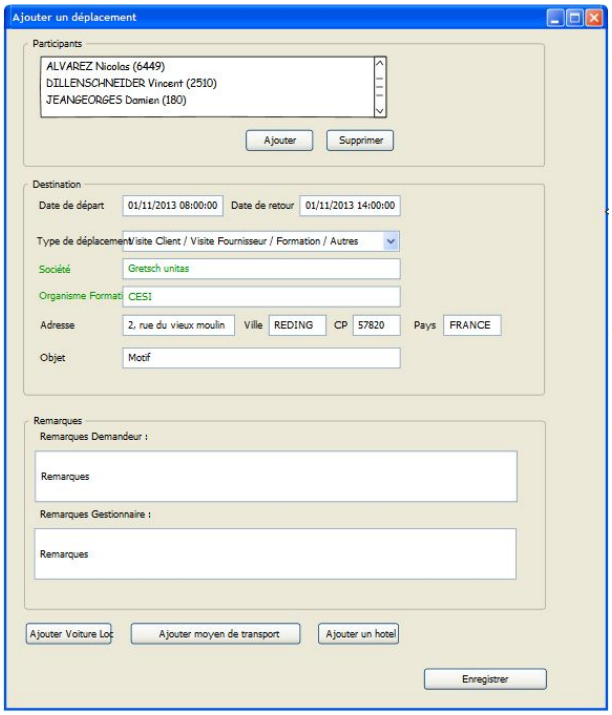


FIGURE 3.12 – Maquette de la saisie d'un déplacement

On peut remarque dans la figure 3.12 que certains champs vont disposer de contraintes spécifiques :

- La date de départ et la date de retour sont les données de temps qui vont être transférées dans le système SAP-HR.
- Le champ société apparaît uniquement si dans le type de déplacement on retrouve "Visite client" ou "Visite fournisseur".
- Le champ organisme apparaît uniquement si dans le type de déplacement on retrouve "Formation".
- Dans les autres cas, ces deux champs sont masqués.

Enregistrer un moyen de transport

Un déplacement est constitué de moyens de transport. Il est à noter qu'il est possible d'avoir plusieurs moyens de transport pour un déplacement. Chaque transport est considéré comme une étape au sein du déplacement. Par exemple, si un commercial effectue le trajet Sarrebourg-Pékin, il pourrait y avoir trois étapes :

- Une voiture de location pour l'étape Sarrebourg-Strasbourg,
- Le train pour l'étape Strasbourg-Paris,
- L'avion pour Paris-Pékin.

La maquette de la figure 3.13 montre les informations à saisir lors de l'ajout d'un moyen de transport.

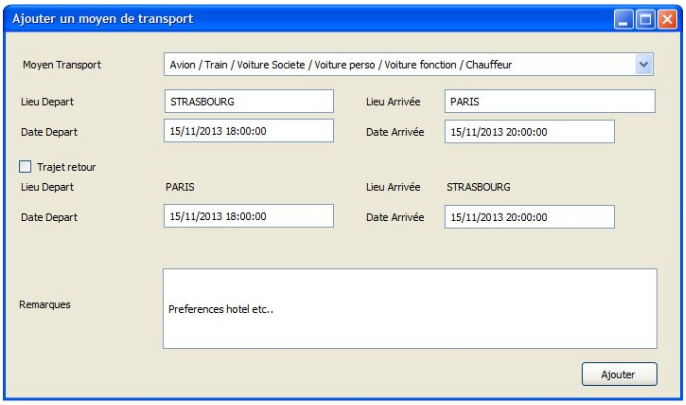


FIGURE 3.13 – Maquette de l'enregistrement d'un moyen de transport

Certains points sont à noter dans cette maquette de la figure 3.13 :
— Le trajet retour sera à renseigner uniquement si la case "trajet retour" est cochée.
— Les villes de départ et d'arrivée du trajet retour seront automatiquement déterminées et affichées par le système en fonction du trajet aller.

Enregistrer une voiture de location

Une voiture de location est un moyen de transport particulier, en effet ce type de transport possède des données qui ne sont pas partagées avec les autres moyens de transport. En dehors de cet aspect, de la même manière que les autres types de transport, ce dernier peut être ajouté plusieurs fois lors d'un déplacement.

La maquette de la figure 3.14 montre les informations à saisir lors de l'enregistrement d'une voiture de location.

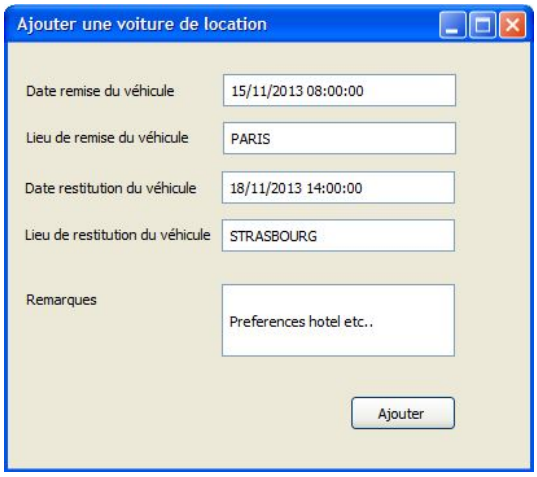


FIGURE 3.14 – Maquette de l'enregistrement d'une voiture de location

Enregistrer un hôtel

Un déplacement peut impliquer la réservation d'un ou plusieurs hôtels. Pour demander la réservation d'un hôtel, le demandeur renseigne le lieu et les dates de départ et d'arrivée, il peut également indiquer des préférences pour la sélection de l'hôtel dans un champ "Remarques" prévu à cet effet.
La maquette de la figure 3.15 montre les informations à saisir lors de la création d'un hôtel.

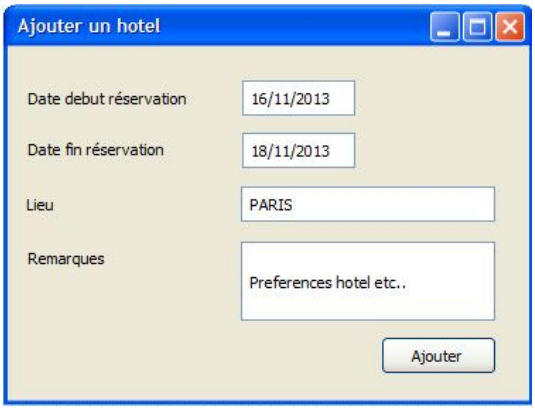


FIGURE 3.15 – Maquette de l'enregistrement d'un hôtel

Traiter les tâches du gestionnaire

Chaque moyen de transport (y compris la voiture de location) et chaque réservation d'hôtel génère une tâche à traiter par le gestionnaire. Dans une interface dédiée, le gestionnaire pourra déclarer les tâches qui ont déjà été effectuées, de cette manière le gestionnaire tout comme le demandeur pourront consulter l'état d'avancement des tâches liées au déplacement.
La maquette de la figure 3.16 montre les informations à saisir lors du traitement des tâches du gestionnaire.

FIGURE 3.16 – Maquette du traitement des tâches du gestionnaire

Rechercher les déplacements

Une interface de recherche devra être disponible afin que les utilisateurs puissent afficher l'histo-

rique des déplacements du système. La maquette de la figure 3.17 montre un exemple d'interface graphique pour que l'utilisateur procède à une recherche des déplacements dans le système.

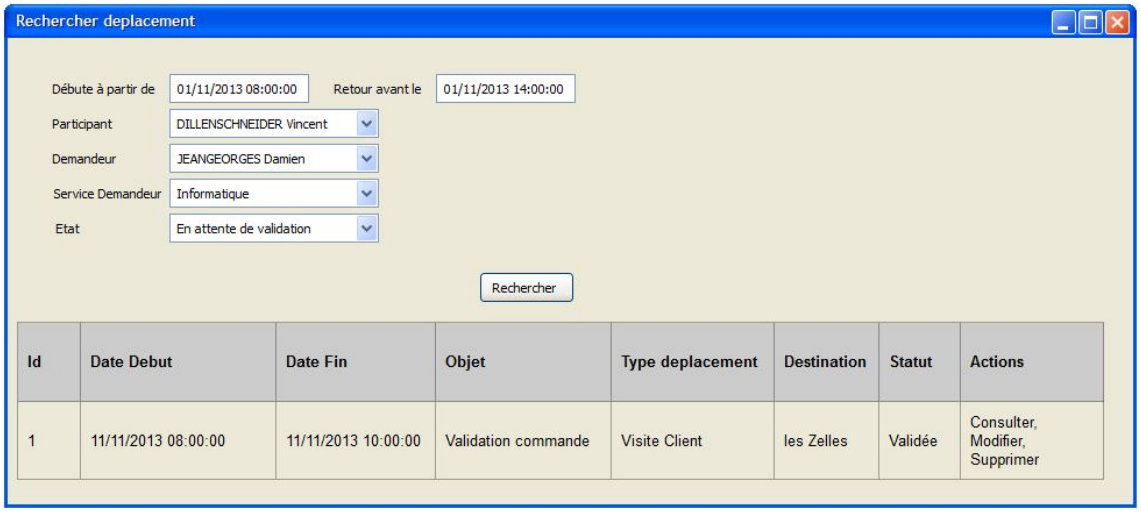


FIGURE 3.17 – Maquette de l'interface de recherche

3.3.2 Etat de l'art

3.3.2.1 Les langages de programmation

La réalisation d'une application informatique s'effectue à l'aide d'un langage de programmation. Il existe un éventail très vaste de différents langages de programmation, et chacun offre différentes possibilités. Les langages de programmation supportent un ou plusieurs styles de programmation (paradigme), parmi ces paradigmes, on peut citer :

- le paradigme impératif (ou procédural) : ce style est basé sur une exécution étape par étape. Une procédure principale (point d'entrée du programme) et des structures permettent de contrôler l'ordre d'exécution des instructions. Le COBOL et le C sont des exemples de langages de programmation qui utilisent ce paradigme.
- le paradigme fonctionnel : ce style est basé sur l'idée d'évaluer une formule et d'utiliser le résultat pour une autre expression. L'ensemble des traitements est effectué en évaluant des expressions et en appelant des fonctions, en effet l'exécution impératif est impossible dans le cadre du paradigme fonctionnel. Lisp est un exemple de langage fonctionnel.
- le paradigme orienté objet : la programmation orientée objet est construit sur le principe que tout est objet. Un objet peut contenir des attributs et des fonctions en rapport avec son sujet. L'objectif du langage orienté objet est de faciliter le découplage et la maintenance d'une application en évitant la redondance de code et en facilitant la réutilisation du code. Ce paradigme permet également d'augmenter la sécurité des programmes en donnant la possibilité de donner une visibilité aux attributs et aux fonctions de l'objet (publique, protégée et privée). JAVA et C++ sont des exemples de langage de programmation orienté objet.

Hormis ces paradigmes, on peut également classer les langages de programmation dans plusieurs catégories :

- les langages compilés : un programme écrit avec ce type de langage a besoin d'un programme annexe (le compilateur) pour être traduit en un langage cible directement utilisable par la machine : le code source génère à l'aide du compilateur un nouveau fichier qui pourra s'exécuter

de manière autonome (fichier exécutable). La modification des sources du programme nécessite une nouvelle compilation du programme pour pouvoir exécuter la nouvelle version. Les langages C, C++ et le COBOL sont des exemples de langages compilés.
— les langages interprétés : un programme écrit à l'aide d'un langage interprété a besoin d'un interpréteur pour pouvoir s'exécuter. Ce type de langage est réputé pour être moins rapide à l'exécution qu'un programme compilé (en effet ceci est expliqué par l'absence d'interpréteur qui traduit les instructions à la volée dans le cas d'un programme compilé). Toutefois, ce type de langage est plus souple car la modification du programme est directement restitué par l'interpréteur. PHP, JavaScript et PERL sont des exemples de langages interprétés.
— les langages semi-interprétés : il s'agit de l'intermédiaire entre le lanage compilé et interprété. Le code source est compilé dans un pseudo-code (bytecode) qui sera interprété par un programme intermédiaire (une machine virtuelle). Cette approche permet une portabilité des programmes via la machine virtuelle (seul élément qui doit être porté). JAVA et C# sont des exemples de ce type de langage.

3.3.2.2 Le World Wide Web

Le World Wide Web (WWW ou Web) permet d'accéder a des ressources sur Internet à l'aide d'un navigateur Internet qui permet la consultation de pages Web proposées par des sites internet à l'aide de serveurs Web. Le Web débute dans les années 1990, à l'origine les pages Web étaient statiques, il s'agissait uniquement d'informations simples à destination de l'utilisateur : on ne faisait pas de la communication mais de l'information sur le Web. En effet, les entreprises utilisaient ce système comme moyen de publication d'informations. Dans les années 2000, l'évolution technique permet de rendre plus dynamique et interactif les sites internet. Dans cette période, on retrouve les notions de blog, de wiki et des sites de réseaux sociaux comme Facebook ou Myspace. Comme il a été montré dans [PORTENEUVE, 2008], les utilisateurs ne sont plus uniquement consommateurs de l'information mais sont devenus également des contributeurs, à partir de cette période le Web devient un mode de communication, on parle alors d'une évolution du Web : le Web 2.0 (le Web social).

3.3.2.3 Serveur HTTP

Un serveur HTTP (ou serveur Web), est un logiciel qui héberge des ressources et qui va assurer la communication des données entre un poste client et le serveur en gérant des requêtes qui respectent le protocole de communication Hypertext Transfer Protocol (HTTP).
Bien que le serveur HTTP le plus utilisé est Apache HTTP Server (utilisé par environ 55% des sites web selon W3Tech.com (voir annexe)), il existe différentes solutions de serveurs HTTP sur le marché.

Apache HTTP Server

Apache HTTP Server (Apache) est un serveur HTTP écrit dans le langage C et qui se trouve maintenu par la Apache Software Foundation (Apache a été historiquement crée par Apache Group). La première version de ce serveur HTTP a été publiée en Avril 1995. Ce serveur est supporté notamment par Windows, Linux et UNIX. Il dispose également d'une grande communauté et se trouve être le plus répandu sur le Web, une explication de sa popularité est probablement lié à son historique (présent sur le marché depuis 1995) ainsi que sa modularité (en effet des fonctionnalités peuvent être activées ou non par de simples directives).

Apache Tomcat

Apache Tomcat est un serveur HTTP qui est également maintenu par la Apache Software Foundation. Il est écrit en JAVA et fournit aussi un conteneur de servlets et JSP pour Java EE.

Internet Information Services

Internet Information Services (IIS) est le logiciel de serveur de services Web développé par Microsoft. Ce logiciel est disponible sur les environnements Windows et permet notamment le développement de sites Web à travers le langage ASP.NET.

3.3.2.4 Système de gestion de base de données

Un système de gestion de base de données (SGBD) est un logiciel qui va permettre de stocker les données persistantes des applications Web. Il existe de nombreuses solutions qui disposent chacune d'avantages et inconvénients.

MySQL

MySQL est un des SGBD les plus utilisés dans le monde. En dehors du milieu professionnel, il est également très utilisé dans le grand public (de par sa gratuité et son historique qui est lié à PHP et aux applications Web). MySQL fonctionne sur de nombreuses plates-formes (notamment sur Windows, Linux et Mac OS X). Ce SGBD a été acheté en 2008 par la société Sun Microsystems, puis a été acquis en 2009 par Oracle Corporation, cette société est propriétaire depuis ce rachat de deux SGBD concurrents : Oracle Database et MySQL.

Microsoft SQL Server

Microsoft SQL Server (MSSQL) est la solution propriétaire de Microsoft qui a été réalisée dans le but d'héberger les données persistantes des applications. On peut citer le fait que ce système ne fonctionne que sur des environnements Windows et peut entraîner des coûts de licences élevés. Toutefois, une annonce a été faite par Microsoft sur la possibilité d'utiliser Microsoft SQL Server sur Linux en 2017 (ce qui confirme le changement de politique de Microsoft concernant sa vision des logiciels libres et le début de son ouverture au monde opensource).

Oracle Database

Oracle Database est un SGBD complet avec une très bonne réputation, il est cependant plutôt réservé aux professionnels de par son coût d'utilisation (même si une version limitée gratuite existe) et sa complexité de mise en oeuvre.

SQLite

SQLite est un SGBD très simple à mettre en place, en effet contrairement aux autres solutions, un unique fichier permet de contenir l'ensemble des informations de la base de données, cependant cette simplicité à un coût : la performance. Même si il est possible d'utiliser SQLite dans les applications Web, on utilisera plutôt cette solution sur des systèmes embarqués, sur des applications Web en test ou encore avec des applications disposant d'une faible charge.

3.3.2.5 Les langages du Web

Dans le but de réaliser une application Web, de nombreux langages et scripts sont à maitriser. Nous allons présentés ci-après, une liste non exhaustive de ces derniers.

PHP

PHP est un langage de script principalement utilisé côté serveur, ce langage est une alternative à Java ou .NET pour produire des pages Web dynamiques via un serveur HTTP. Il s'agit d'un langage interprété : le serveur HTTP interprète le code PHP de l'application et va générer un résultat (code HTML, données binaires, etc.) qui est ensuite rendu par le navigateur du poste client. Concernant le serveur utilisé, il s'agit fréquemment d'un serveur Apache de par sa gratuité et sa portabilité, il est en effet possible de l'installer sur un environnement Linux ou Windows (il est aussi possible d'utiliser la solution propriétaire IIS de Microsoft comme serveur HTTP pour PHP).

Ce langage est souple et peu typé, il est par conséquent plus facile à prendre en main qu'un langage plus strict tel que Java qui impose une rigueur plus élevée. En effet, PHP permet d'obtenir un résultat sans écrire un code source volumineux et verbeux, c'est probablement un des aspects qui explique sa popularité. PHP est très souvent utilisé pour la réalisation des sites internet (81,7% selon W3Techs.com, voir annexe) pour plusieurs raisons :

- son apprentissage est rapide,
- une grande communauté et de nombreuses ressources sont disponibles,
- il existe une grande variété de projets opensource pour ce langage,
- de nombreux Content Management System (CMS) sont disponibles sur ce langage,
- le support du langage PHP est disponible quasiment sur l'ensemble des hébergeurs Web, contrairement à d'autres langages comme Python ou Java.

PHP n'arrête pas son évolution : il permet depuis sa version 5, la prise en charge de la programmation orientée objet, la version 5.3 a apporté le support des espaces de noms (namespaces), ce qui permet une plus grande souplesse dans l'utilisation des noms des classes.

ASP.NET

ASP.NET est la technologie propriétaire de Microsoft qui est une évolution majeure de ASP (Active Server Pages), elle se trouve maintenant incorporée dans la plate-forme Microsoft .NET. Cette technologie repose sur le service Web Internet Infomration Services (IIS) de Microsoft. ASP.NET est utilisé pour mettre en oeuvre des applications Web dans un environnement Microsoft "natif". Le développement d'applications Web sur cette technologie peut être réalisé avec plusieurs langages de programmation que propose la plate-forme .NET : Visual Basic .NET, C#, JScript, etc.

JEE

La plate-forme Java EE (Java Enterprise Edition, anciennement J2EE) est un ensemble de spécifications destinées aux applications Web d'entreprise. Tout comme le langage de programmation java, Java EE est également maintenue par la société Oracle. Cette plate-forme a pour objectif de faciliter le développement d'applications Web, en proposant un ensemble d'API visant à répondre aux problèmes les plus couramment rencontrés par les développeurs. Ces derniers restent libres d'utiliser totalement ou partiellement les spécifications de Java EE. La plate-forme JEE a été pensée de manière à délivrer des solutions pour le support des applications « critiques » de l'entreprise (sécurité, montée en charge, performances, etc.) [GROUSSARD, 2011].

HTML5

HyperText Markup Language 5 (HTML5) est la dernière version de HTML. HTML est un langage de présentation (à l'aide de balises) de l'information qui est utilisé par les pages Web. Cette dernière version du langage a été terminée le 28 octobre 2014 et apporte un nombre de nouveautés et d'évolutions importantes par rapport à HTML4 (liste non exhaustive) :

— un allègement du code : des balises ont été simplifiées,
— de nouvelles balises sémantiques qui permettent de mieux structurer et analyser les pages,
— la gestion du multimedia, il n'y a plus besoin de plugin externe pour lire des vidéos, une balise intègre directement un lecteur vidéo ou audio dans la page Web,
— des possibilités pour le développement de jeux, la réalisation d'animations et de rendu 3D à l'aide des balises canvas,
— des formulaires améliorés où il est possible d'obtenir des contrôles de validation du contenu avant l'envoi du formulaire au serveur Web.

Les navigateurs Internet modernes supportent de nombreuses fonctionnalités de HTML5, toutefois il est judicieux de suivre la recommandation de [RIMELE, 2011] et de vérifier la compatibilité des navigateurs via un site tel que [html5test].

CSS3

Cascading Style Sheets (CSS) est un langage qui décrit la présentation des documents HTML et XML. Le navigateur Internet prend en charge les instructions CSS pour effectuer le rendu des pages Web. CSS3 est une évolution de la version CSS 2.1 et apporte un ensemble de nouveautés, dont notamment :

— de nouveaux sélecteurs,
— de nouveaux effets pour les éléments HTML,
— une détection des caractéristiques de l'appareil de l'utilisateur et l'adaptation des instructions à l'aide des media queries.

JavaScript

JavaScript est un langage de programmation de scripts utilisé principalement dans les pages Web. Ce langage créé dans les années 1990 se trouve maintenant être un élément incontournable lors de la réalisation de sites internet dynamiques. Au sein d'une page HTML, il est possible avec JavaScript de manipuler les événements de la page Web, de modifier les éléments HTML ou encore de dialoguer avec le serveur Web à l'aide de requêtes AJAX.

3.3.2.6 Les frameworks

Un framework est un ensemble cohérent de composants logiciels qui permettent de répondre à une problématique précise, par exemple un framework pour la gestion des données (framework ORM), ou encore un framework spécialisé pour la présentation des données (framework front-end). L'utilisation d'un framework impose un cadre de travail et donc de respecter l'architecture que peut imposer le framework.

Framework ORM

L'object-relational mapping (ORM) est une technique qui consiste à donner la possibilité aux développeurs de travailler « virtuellement » avec une base de données orientée objet à partir d'une base de données relationnelle. Ceci est possible en définissant, et en configurant des correspondances

entre la base de données et les objets de l'application. Finalement, les développeurs ne vont plus utiliser le langage SQL directement, ils manipuleront à la place, les objets qui vont reproduire les modifications dans la base de données. Il existe plusieurs frameworks en fonction des langages de programmation, on peut citer notamment Doctrine ou Propel pour PHP, Hibernate ou JPA dans le monde de JAVA et on retrouve également DataObjects.NET ou Entity Framework pour la plate-forme .NET.

Framework front-end

Un framework front-end est un framework qui va faciliter la création des interfaces graphiques pour les clients (PC, smartphones et tablettes). Ce type de framework devrait permettre de rapidement développer des interfaces graphiques avec un rendu de qualité. L'utilisation d'un framework front-end offre également un avantage très important : la compatibilité des navigateurs. En effet, devoir adapter les applications pour que le rendu soit correct sur la majorité des navigateurs Web du marché est un des problèmes récurrents du développeur Web. Attention cependant, car l'utilisation d'un framework Web peut également contraindre le choix des navigateurs. Il est effectivement courant de retrouver des soucis de compatibilité avec des anciens navigateurs tel que Internet Explorer en dessous de sa version 9. Plusieurs frameworks front-end sont disponibles sur le marché dont notamment Bootstrap, Foundation ou encore Semantic UI.

Framework Web MVC

Un framework Web MVC (Modèle-Vue-Contrôleur) va permettre de réaliser une application Web en respectant une architecture MVC, c'est à dire en séparant les différentes couches de l'application. L'objectif principal de ce type de framework est avant tout de simplifier les tâches dans la réalisation d'une application Web. En effet, l'utilisation d'un framework va permettre d'utiliser des composants déjà testés et de ce fait permettra au développeur de ne pas réinventer un code existant et de se concentrer sur uniquement sur les besoins métiers à couvrir. Pour autant, il n'est pas forcément nécessaire d'utiliser systématiquement un framework MVC, en effet il faut prendre en compte certains éléments avant de se décider d'utiliser un framework : la taille du projet, les contraintes techniques et le temps d'apprentissage du framework sont notamment des paramètres à prendre en compte. Il existe différents frameworks Web MVC, on retrouve notamment Symfony 2 ou ZendFramework 2 pour PHP, Java EE ou Spring pour Java et il existe également ASP.NET MVC ou DotnetNuke pour la technologie ASP.NET.

3.3.2.7 Les moteurs de template

Un moteur de template permet d'améliorer la lisibilité du code de la couche de présentation et permet également de séparer nettement la couche d'interface graphique de l'application. Ce point est un avantage important car cela permet également de séparer les compétences de l'équipe, en effet un expert dans le design de l'application n'a pas besoin de connaître un langage de programmation précis pour intervenir sur la couche de présentation de l'application. De plus, un moteur de template permet d'éviter de la redondance de code en permettant de structurer les pages par inclusion ou héritage.

La figure 3.18 illustre ce principe en présentant la différence entre un affichage de pages Web sans moteur de template et un affichage avec l'utilisation d'un moteur de template. La partie en couleur rouge de la figure représente la redondance de code.

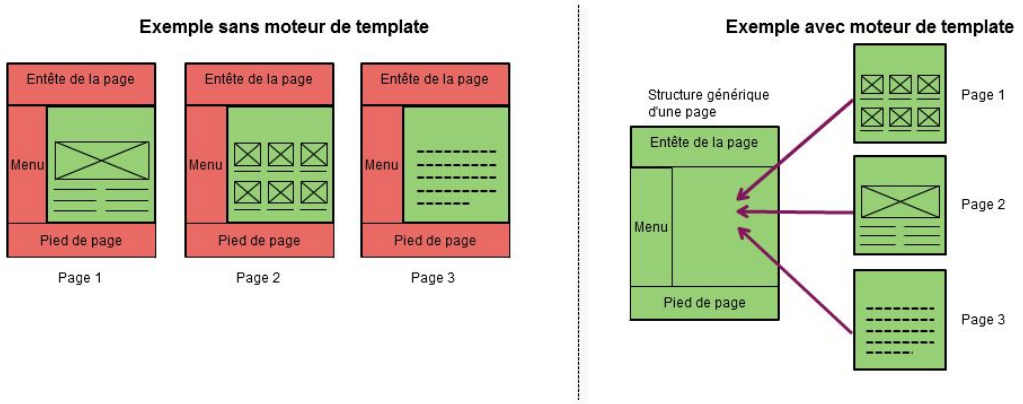


FIGURE 3.18 – Utilisation d'un moteur de template

Les moteurs de template ne sont pas réservés à un seul langage de programmation, on trouve des solutions sur JAVA (JSF, FreeMarker), PHP (Twig, Smarty) ou encore sur la plate-forme .NET (String template, ASP.NET Razor).

3.3.2.8 Architecture d'une application

Avec l'évolution de l'informatique, plusieurs modèles d'architecture dans le développement informatique ont vu le jour pour répondre à différents besoins (application bureautique, architecture client-serveur, etc.). Parmi ces architectures, on retrouve : l'architecture 1-tier, l'architecture 2-tier, l'architecture 3-tiers et l'architecture n-tiers.

L'architecture 1-tier

Il s'agit de la plus simple forme d'architecture, elle peut se résumer à une application qui fonctionne de manière totalement indépendante et où l'ensemble de ses éléments se retrouve sur le même environnement sans découplage (par exemple une application bureautique).

La figure 3.19 illustre une architecture où l'ensemble des couches d'une application repose sur le même environnement (architecture 1-tier).

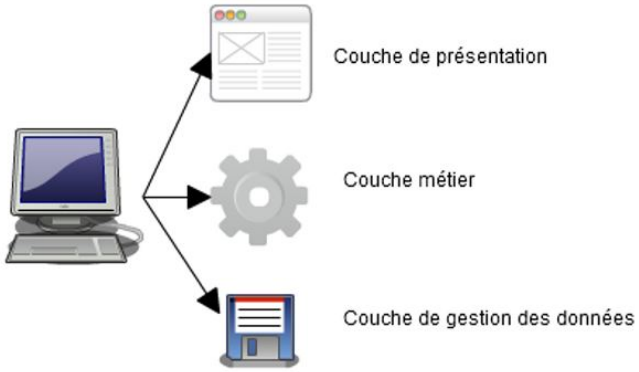


FIGURE 3.19 – Architecture 1-tier

L'architecture 2-tiers

L'architecture 2-tiers (aussi appelé architecture client-serveur) est le résultat de la séparation de l'application en deux parties (ou couches) entre un client et un serveur. On peut citer deux modèles dans cette catégorie d'architecture, le premier modèle est celui que l'on nomme "client lourd" où la présentation et la logique de traitement sont réalisées par le client alors que la couche de données est réalisé par le serveur. L'avantage de ce premier modèle est d'augmenter l'expérience utilisateur en lui proposant une interface riche, cependant les inconvénients de ce modèle sont les difficultés de maintenance du client et la grande charge d'opérations qui peuvent reposer sur ce dernier.

La figure 3.20 illustre le modèle du client lourd.

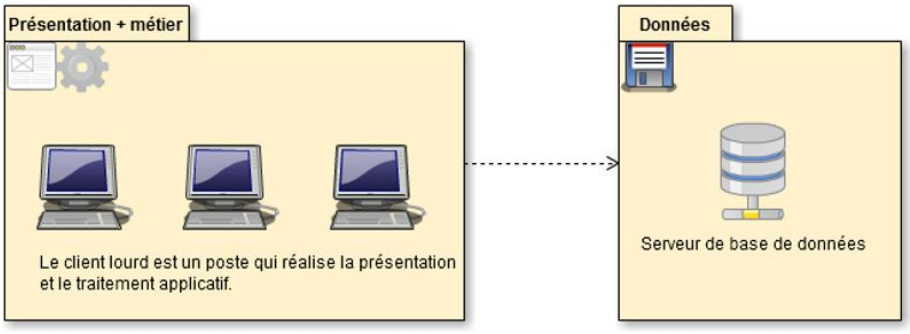


FIGURE 3.20 – Client lourd

Le second modèle est celui du "client léger", dans ce modèle la présentation est effectuée par le client, alors que la logique de traitement et la couche de données sont réalisées par un serveur. Les principaux avantages de ce modèle sont les facilités de maintenance et de déploiement (pas de configuration spéciale à avoir au niveau du client), toutefois une montée en charge du serveur peut être problématique dans cette configuration.

La figure 3.21 illustre le modèle du client léger.

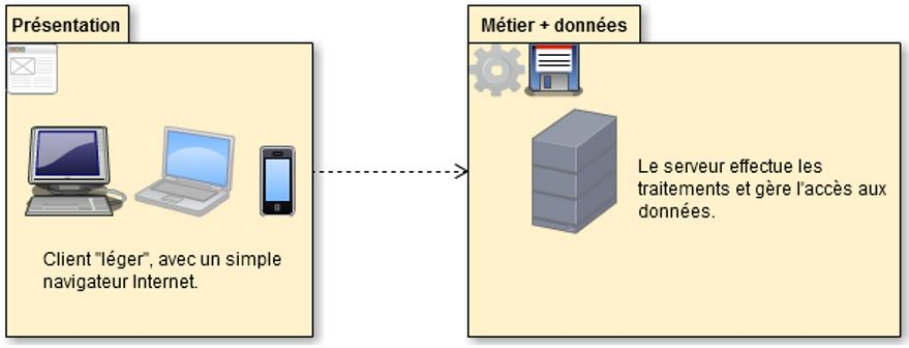


FIGURE 3.21 – Client léger

L'architecture 3-tiers

L'architecture 3-tiers est une extension du modèle client-serveur vu précédemment et il se trouve être le cas le plus simple et le plus utilisé de l'architecture n-tiers.

Au sein de ce modèle, on retrouve trois éléments qui possèdent chacun une responsabilité bien défini. En premier lieu, on retrouve un client qui est constitué d'un simple navigateur Internet et dont le rôle est uniquement de s'occuper de la présentation de l'application. Le deuxième élément est un serveur applicatif, qui va prendre en charge la partie logique de l'application en effectuant les opérations et les traitements métiers. Enfin, le dernier élément est un serveur de base de données qui va prendre en charge la persistance et les accès aux données. Ce modèle possède tous les avantages de l'architecture client-serveur, tout en améliorant la maintenance avec un découplage de toutes les couches d'une application. De plus, ce modèle apporte aussi plus de flexibilité, il est en effet possible de modifier un élément (le serveur de base de données par exemple) sans impacter l'affichage ou le serveur applicatif. Cependant réaliser une architecture 3-tiers est souvent plus élevé en terme de coûts qu'une architecture 2-tiers, ceci est justifié par une plus grande expertise nécessaire pour la réalisation de ce type d'architecture.

La figure 3.22 illustre l'architecture 3-tiers.

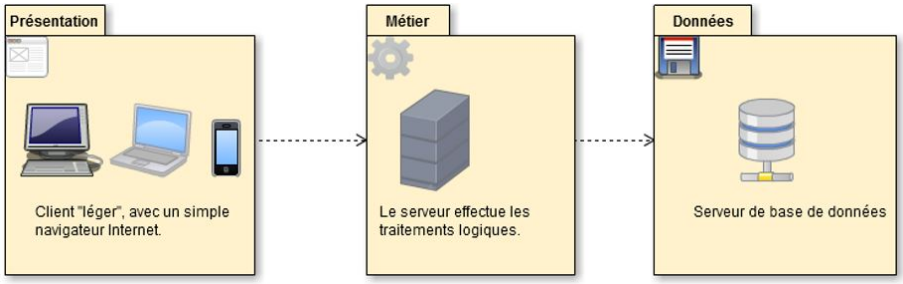


FIGURE 3.22 – Architecture 3-tiers

L'architecture n-tiers

L'architecture n-tiers (ou encore architecture distribuée) est une précision de l'architecture 3-tiers, où une couche est affinée pour augmenter la maintenabilité et la robustesse de l'application. Ces gains sont réalisés à l'aide d'un meilleur découplage, via des interfaces et un nombre de tiers plus élevé.

L'architecture n-tiers permet de séparer clairement l'ensemble de l'application, elle offre également de grandes possibilités d'extension et de flexibilité, il est en effet possible à l'aide des interfaces de modifier un composant d'une application (une sous-couche) sans avoir à se soucier de l'impact sur l'ensemble de l'application (à condition évidemment que les interfaces sont respectées), toutefois son développement est encore plus coûteux que l'architecture 3-tiers.

La figure 3.23 montre un exemple d'architecture n-tiers.

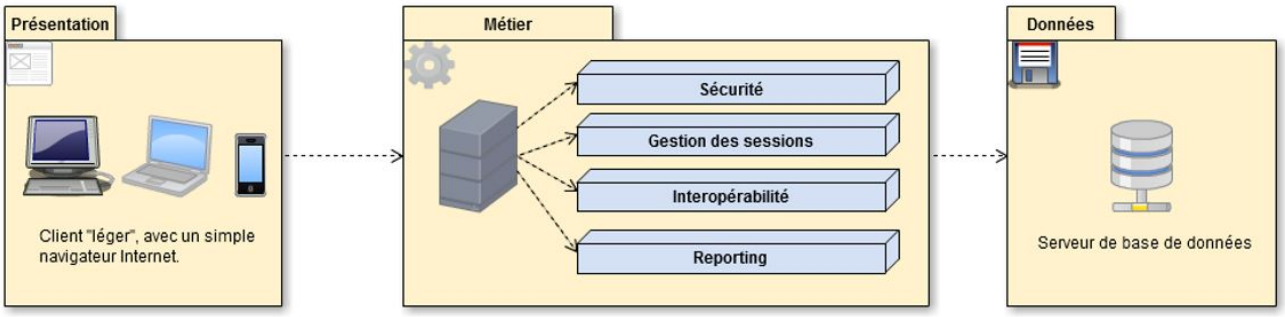


FIGURE 3.23 – Architecture n-tiers

3.3.2.9 Communication client-serveur

De nombreuses technologies de communication client-serveur existent pour réaliser des applications Web. On peut différencier néanmoins la technologie servant à la communication (SOAP, AJAX, REST, XML-RPC) et le format de données qui est utilisé pour la transmission des données (XML, JSON). Les services mis à disposition par le serveur à travers le protocole HTTP (services Web) peuvent être protégés par un protocole qui permet d'appliquer de la sécurité aux services Web : le WS-Security.

Les formats de données

Différents formats de données existent et ces derniers permettent de formaliser la transmission des données.

XML L'Extensible Markup Language (XML) est un langage de structuration de l'information à l'aide de balises extensibles. XML est un format de description des données et la mise en forme de ces données est assurée par un langage additionnel : CSS, XSL et XSLT (permet de transformer un document XML en un document HTML ou PDF par exemple). Son utilisation est particulièrement adaptée pour l'échange d'informations, en effet la conformité des fichiers et la cohérences des données peuvent être validées à l'aide des schémas XSD (et plus anciennement à l'aide des DTD).

JSON JavaScript Object Notation (JSON) est un format léger d'échange de données, il est dérivé de la notation des objets du langage JavaScript. Il est facile à lire ou à écrire pour des humains et il est simple à analyser par les ordinateurs. JSON est un format complètement indépendant de tout langage et des parsers (outils qui permettent l'analyse du JSON) existent dans la majorité des langages de programmation. Parmi les avantages de ce format, on retrouve les points suivants :

- peu verbeux, ce qui donne un fichier moins volumineux que d'autres formats et donc une économie de la bande passante,
- facile à apprendre,
- simple à parser.

Outre la communication client-serveur, ce format est aussi utilisé pour la génération de fichiers de configuration, la sérialisation et la désérialisation d'objets.

Les technologies de communication

Les technologies qui permettent de réaliser la communication entre le client et le serveur dans des applications Web qui utilisent le protocole HTTP comme support de diffusion.

REST Representational State Transfer (REST) est un style d'architecture permettant de construire des applications Web. Ce n'est pas une technologie à part entière mais plutôt un ensemble de bonnes pratiques et de conventions. L'architecture REST propose le fait que le protocole HTTP est suffisant pour couvrir l'ensemble des besoins d'un service Web. L'accès et la gestion d'une ressource se fera à l'aide d'une adresse HTTP, par exemple : *http ://monapplication.fr/users/14* va permettre de traiter l'objet users avec l'identifiant 14.

Les opérations basiques proposées par le protocole HTTP sont alors utilisées pour interagir avec l'objet, on retrouve pour une ressource les opérations suivantes [PORTENEUVE, 2008] :
- Créer (create) => POST,
- Afficher 'read) => GET,
- Mettre à jour (update) => PUT,
- Supprimer (delete) => DELETE.

Afin d'illustrer ces méthodes, en reprenant l'objet users ci-dessus on pourrait avoir comme opérations :
- *GET http ://monapplication.fr/users/14*, permet de récupérer les informations de l'utilisateur 14
- *POST http ://monapplication.fr/users*, permet de créer un nouvel utilisateur
- *PUT http ://monapplication.fr/users/14*, permet de mettre à jour l'utilisateur 14
- *DELETE http ://monapplication.fr/users/14*, permet de supprimer l'utilisateur 14.

Le format de la transmission des informations peut être réalisé en XML ou en JSON.

AJAX Asynchronous JavaScript and XML (AJAX) est une technologie qui permet de rendre dynamique des applications Web. Avant l'utilisation de cette technologie, un utilisateur devait rafraîchir intégralement une page Web pour mettre à jour son contenu, maintenant il est possible de rafraîchir une partie de la page Web sans rechargement visible par l'utilisateur. Le point précédent est important pour plusieurs raisons :
- l'économie de la bande passante qui est réalisé, en effet en rechargeant un élément de la page plutôt que son intégralité le réseau est moins sollicité,
- les possibilités supplémentaires de rendre dynamique des champs de saisie (autocomplétion par exemple) depuis des informations du serveur,
- des contrôles dynamiques sur des champs de saisie, sans avoir à envoyer l'intégralité du formulaire au serveur Web,
- l'augmentation de la réactivité de l'interface utilisateur,
- la réduction du temps de latence,
- la satisfaction de l'utilisateur, il est effectivement plus agréable d'avoir uniquement du contenu qui se met à jour sans avoir à réactualiser l'ensemble d'une page.

L'utilisation d'AJAX est tout à fait recommandé pour effectuer du transfert de données rapides entre un client et un serveur. Pour réaliser ce point, AJAX va utiliser l'objet XMLHttpRequest qui va servir au dialogue asynchrone avec le serveur et va s'appuyer sur des formats de données pour transférer les informations. En dehors du format XML, il est possible d'utiliser d'autres formats de données tels que JSON. Pour autant AJAX n'est pas forcément adapté à toutes les utilisations, il est généralement plus compliqué de mettre en oeuvre cette technologie plutôt que d'utiliser une approche classique de développement.

La sécurité des services Web : WS-Security

WS-Security (WSSE) est une méthode d'authentification disponible pour les services Web. Comme le protocole HTTP est sans état, sécuriser des services Web est compliqué, WSSE fournit une méthode pour répondre à ce problème. En respectant cette spécification, lorsque le client génère une requête HTTP, il accompagne cette dernière avec un header spécifique qui sera utilisé uniquement pour l'authentification du client.

La figure 3.24 illustre cette méthode d'authentification.

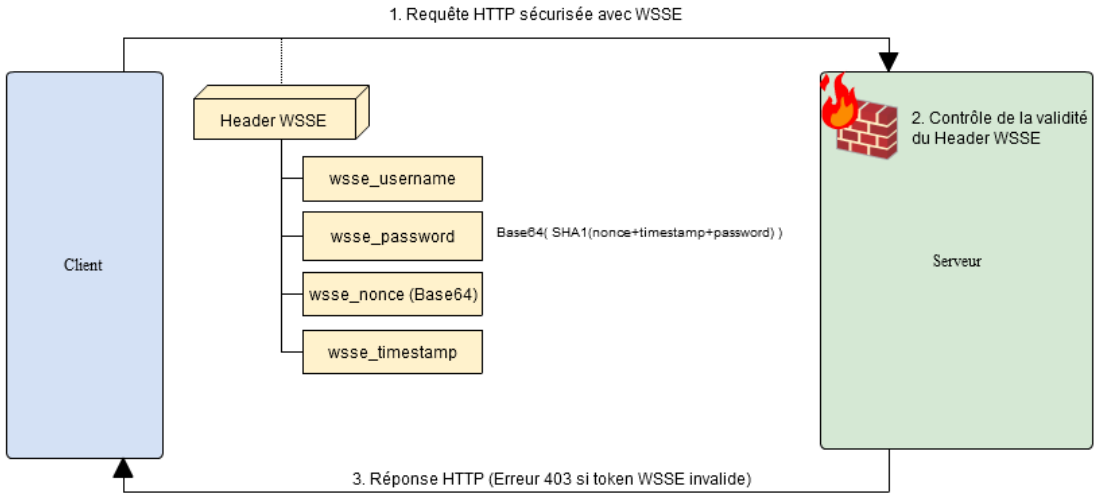


FIGURE 3.24 – Méthode d'authentification WSSE

Au sein de ce header nous allons retrouver :
— l'identifiant du client (wsse_username).
— un nonce (wsse_nonce), un élément utilisé dans la génération du mot de passe qui va permettre de renforcer sa sécurité.
— le timestamp (wsse_timestamp) qui est la date de création utilisée lors de la génération de la requête.
— le mot de passe (wsse_password), qui sera le produit encodé en SHA1 du mot de passe initiale, avec le nonce et le timestamp.

Le serveur va de son côté pour chaque requête HTTP entrante vérifier la validité de la requête en procédant aux actions suivantes :

1. Vérifier si le timestamp est > à X minutes, ceci afin d'empêcher de relancer une requête identique.
2. Vérifier si le nonce a déjà été utilisé (par exemple en stockant les nonces utilisés les X dernières minutes dans le cache du serveur).
3. Vérifier enfin si la génération du mot de passe du côté serveur (avec les éléments fournis par le client) est identique à celui généré par le client. Si oui la requête est valide, dans le cas contraire une erreur 403 est envoyée au client.

3.3.2.10 Environnement de développement intégré

Un environnement de développement intégré (EDI ou IDE en anglais) offre des outils pour faciliter le développement d'application informatique, ceci afin d'augmenter la productivité des développeurs. Il est effectivement impensable aujourd'hui de développer une application professionnelle complexe qui va nécessiter l'édition de plusieurs dizaines ou centaines de fichiers à l'aide d'un simple éditeur de texte. Parmi les fonctionnalités "standards" que l'on retrouve dans un EDI, on peut citer notamment :

- l'auto complétion des variables et des fonctions en rapport du contexte dans lequel on se trouve : méthodes de l'objet, fonctions accessibles, etc...
- l'import des bibliothèques et packages nécessaires au bon fonctionnement du programme,
- une vue qui permet d'inspecter le fichier courant, ce qui permet d'avoir rapidement un aperçu des fonctions et variables déclarées dans le fichier,
- l'intégration des systèmes de gestion de versions,
- un outil intégré pour réaliser les tests,
- une documentation accessible dans l'éditeur pour le langage utilisé.

Un environnement de développement intégré peut être multilangage ou dédié à un langage précis. Eclipse et Netbeans sont des IDE populaires capables de prendre en charge un nombre important de langages de programmation (Java, PHP, C++, etc.).

3.3.2.11 Gestion des versions

Un système de gestion de versions offre la possibilité de travailler en équipe sur un projet et permet également de gérer la concurrence des modifications sur les fichiers. En dehors de ces aspects collaboratifs, un système de gestion de versions améliore grandement la sécurité du projet en permettant d'archiver les modifications des fichiers, de revenir sur une ancienne version, ou encore de gérer des modifications conséquentes du projet à l'aide de "branches" sans affecter le développement principal. Le travail peut alors être réparti entre plusieurs personnes qui vont devoir valider leurs modifications et éventuellement gérer les conflits survenus si des modifications concurrentes sont arrivées sur la même portion de fichier.

Un tel système apporte vraiment un énorme avantage dans la gestion du code source du projet. Toutefois, la décision de réaliser un projet avec l'aide d'un outil de gestion de versions n'est pas sans conséquence dans le processus de développement, en effet l'équipe doit d'abord définir comment utiliser ce système dans le cadre du projet : Comment gérer les branches ? Comment numéroter les versions ? Quand publier ses modifications à l'équipe ?

Plusieurs systèmes de gestion de versions existent, parmi les solutions disponibles "GIT" est devenu très populaire car son fonctionnement est simple tout en apportant des fonctionnalités très avancées. De plus son intégration avec les IDE est très répandue.

3.3.3 Les contraintes de l'environnement

3.3.3.1 Contraintes organisationnelles

Ressources humaines

L'ensemble du projet sera étudié et réalisé par l'équipe NTIC de la société FERCO. D'autres acteurs seront amenés à intervenir au niveau du projet, notamment l'équipe infrastructure qui devra fournir

un support technique au niveau des serveurs.

Délais

Le planning prévisionnel du projet est de neuf mois et doit débuter en Août 2013.

3.3.3.2 Contraintes budgétaires

Aucun nouvel investissement et aucune prestation externe ne sont à prévoir. Les seules éléments budgétaires sont les ressources internes mobilisés (matériel et ressources humaines).

3.4 Planification

La figure 3.25 illustre l'ordonnancement des tâches du projet.

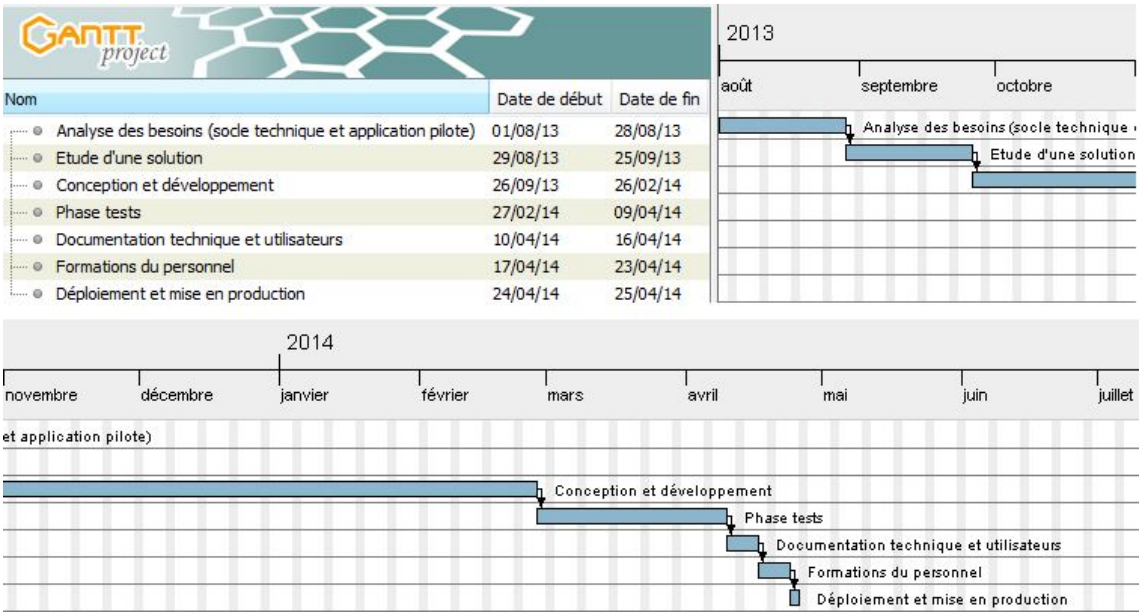


FIGURE 3.25 – Phases du projet

4 Conception, réalisation et test

4.1 Choix technologiques

4.1.1 Le langage

De nombreux langages existent pour développer des applications Web, cependant PHP est le langage qui a été rapidement retenu et cela pour plusieurs raisons. Tout d'abord PHP est un langage très utilisé dans le développement Web et dispose d'une large communauté d'utilisateurs et de nombreux produits (librairies et framework) opensource qui viennent enrichir ce langage, il serait donc aussi facile pour la société de trouver des ressources externes maîtrisant ce langage en cas soit de changement de politique qui viserait à externaliser cette activité de développement ou au contraire d'embaucher une personne avec ces compétences pour renforcer l'équipe interne. Un autre point à ne pas négliger est le fait que l'équipe en charge du développement du projet dispose de fortes compétences sur ce langage, cela permettra logiquement de réduire le temps de formation nécessaire à uniquement la prise en main de nouveaux framework, cela permettra aussi par conséquent à réduire les coûts du projet. Et enfin, comme migrer l'intégralité de l'ancien PAE vers le nouveau est impossible dans le cadre de ce projet en raison de la charge de travail trop importante que cela représenterait, une cohabitation des deux PAE aura lieue pendant une durée encore indéfinie, il est donc plus facile dans le cadre de la maintenance d'intervenir sur le même environnement (langage et serveur), ce qui renforce la décision du choix du langage PHP.

4.1.2 La base de données

La société Ferco dispose déjà de nombreuses applications au sein de son SI sous différents SGBD (MySQL, PostgreSQL et Microsoft SQL Server). Cependant, le seul SGDB géré par le service Informatique et qui dispose d'un paysage système plus avancé (serveur productif et tests) se trouve être MySQL. En effet, PostgreSQL et Microsoft SQL Server sont utilisés uniquement lorsque une solution d'un éditeur tiers doit être installée et que cette dernière impose son SGBD, c'est alors à l'éditeur en question de mettre en place une organisation pour gérer sa base de données.

La figure 4.1 présente l'architecture actuelle de MySQL dans la société FERCO, dont la sécurité est effectuée à l'aide de réplications et de sauvegardes [SEGUY, 2007].

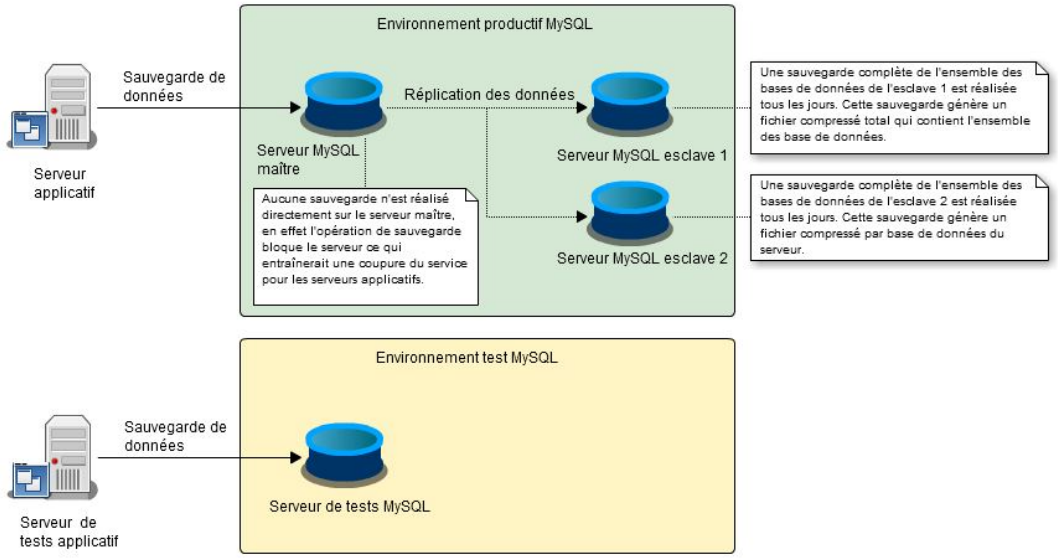


FIGURE 4.1 – Architecture de la solution MySQL à FERCO

Dans le cadre de notre projet, nous avons logiquement opté pour la solution de MySQL afin de respecter l'organisation de la société.

4.1.3 Le framework MVC

Déterminer le framework MVC qui va permettre de réaliser facilement une application avec une architecture MVC est une étape importante car ce dernier va déterminer la base de notre PAE. Utiliser un framework MVC est un choix qui est justifié par l'ampleur du travail à réaliser qui est trop importante afin d'envisager le développement d'une solution complète "maison" par l'équipe interne. De plus en utilisant un framework, un travail conséquent a déjà été réalisé et testé en amont sur les problèmes récurrents que rencontrent les développeurs. Enfin, l'utilisation d'un framework facilitera la reprise de l'application par d'autres développeurs en demandant comme pré-requis des compétences de la personne sur le framework ou le middleware retenu.

Le choix du langage de programmation PHP qui a été défini est un élément technique important, car ce dernier impose déjà d'écarter des solutions comme Java EE, Django (Python) ou encore Ruby on Rails (Ruby). Au sein du langage PHP, on retrouve parmi les frameworks MVC les plus complets et disposant de beaucoup de ressources documentaires : Zend Framework 2 (ZF2) et Symfony 2 (SF2). Notre choix se fera donc entre ces deux framework qui permettent la réalisation d'applications Web.

Choix du framework MVC

Après une première expérience avec ZF1 sur le portail Intranet, il aurait été logique d'aller directement ou de privilégier ZF2. Cependant, nous avons tout de même testé durant notre phase d'étude ces deux frameworks. Nous avons décidé que les critères importants à prendre en compte afin de déterminer la solution retenue seraient la qualité, la documentation disponible et la maturité du produit.

Le premier framework testé a été Symfony 2, afin de procéder au test de ce dernier, nous avons suivi la documentation du site officiel [Symfony] qui était très complète et agréable à lire. Nous avons

été rapidement très surpris par Symfony 2, en effet la philosophie de ce framework est de respecter les bonnes pratiques du développement logiciel et il arrive à tenir ses promesses avec des arguments comme la séparation des couches du logiciel, l'utilisation de design pattern (dont la dépendance d'injection), le paramétrage de l'application dans des fichiers ou annotations de classes, le routage des requêtes intuitif, etc. En dehors de fournir une solution respectueuses des standards, le framework est livré avec un ensemble d'outils qui visent à augmenter la productivité du développeur, on peut citer notamment des commandes disponibles dans le terminal qui permettent d'automatiser un certain nombre de traitements ou encore une barre d'outils permettant l'introspection de l'application, il est important de noter que cette dernière est disponible en standard sur la page Web (lorsque l'application se trouve en mode développement).

Le second framework testé a été ZF2, nous avions noté effectivement que Zend avait totalement refondu son framework, aucune reprise depuis un projet ZF1 n'est donc possible. Dans le cadre de notre projet, la reprise d'un existant n'étant pas d'actualité ce point n'est pas bloquant. Pour tester ce framework, nous avons suivi un tutoriel afin d'initialiser une nouvelle application Web, et c'est à ce moment que nous avons compris que le framework était assez limité par rapport à SF2. Là où nous avons été très surpris par SF2, nous avons été déçus par ZF2 : pas de séparations claires des couches du logiciel, la configuration est réalisée avec du code PHP dans des tableaux et la documentation sur le site officiel est moins complète. Finalement, l'impression que laisse ZF2 est qu'il vise à fournir une riche bibliothèque de composants plutôt que de proposer un ensemble complet d'outils. Mais contrairement à Symfony 2, ZF2 est plus libre dans ses conventions de codage et de hiérarchisation de dossiers.

Malgré beaucoup de promesses sur ZF2, notre choix s'est rapidement porté sur Symfony 2 car ce dernier nous a paru plus mature en apportant des outils adaptés aux développeurs, d'une meilleure qualité en proposant un ensemble cohérent qui vise à respecter les bonnes pratiques et finalement offre une documentation riche tout en ayant une communauté d'utilisateurs plus importante.

Présentation de Symfony 2

Symfony 2 est un framework écrit en PHP, qui va permettre aux développeurs de réaliser un travail de qualité, rapidement et surtout qui sera facile à maintenir par la suite. Symfony 2 est développé par la société Sensio Labs, qui en dehors de ce framework est à l'origine de plusieurs produits opensource (Silex, Twig, Swiftmailer, etc.), Sensio Labs délivre également des formations et des certifications sur Symfony 2.

Symfony 2 est orienté pour effectuer du développement Web, mais il est possible de réaliser d'autres types d'application (par exemple des applications de type console). De par son fonctionnement, il sépare efficacement tous les travaux nécessaires à la création d'un site internet complexe (formulaires Web, interface graphique, gestion de la traduction, etc.). Au niveau des performances, il s'agit d'un framework qui utilise un système de cache efficace qui permet de réduire de façon significative le temps de génération des pages et d'obtenir par conséquent des gains de temps importants. Symfony 2 dispose également d'un système de gestion de la sécurité innovant, ainsi que d'un système de gestion du routage des requêtes très pratique et intuitif. Ces deux systèmes sont configurés via des fichiers de configuration, ce qui entraîne une maintenance de la solution plus simple. Il est par ailleurs parfaitement couplé avec d'autres librairies populaires de la communauté PHP, dont l'excellent Doctrine2 (qui est intégré par défaut dans la distribution de SF2), qui est est un ORM performant disponible pour le langage PHP. Enfin, SF2 fournit une barre d'outil très pratique permettant d'analyser un code, cette barre d'outils fournit un ensemble de services utiles tels que le routage par

contrôleur (plus simplement, cela permet de savoir où se situe précisément le code qui s'exécute : classe, contrôleur et action). Outre ces aspects, SF2 met à disposition du développeur un outil qui permet l'affichage des timings d'exécution, ceci afin de connaître rapidement quelle partie du code pourrait être optimisée, cet outil offre aussi un affichage ergonomique de la pile d'exécution, ainsi que de nombreuses fonctionnalités concernant la gestion de l'identification et de l'authentification et encore d'autres éléments que nous ne citerons pas ici.

Pour résumer, Symfony 2 est un framework Web utilisant l'architecture d'applications MVC, ses atouts sont nombreux, on notera particulièrement les points visant à corriger les limitations de l'ancien PAE : la rapidité, la flexibilité et la possibilité d'utiliser des composants fiables et réutilisables afin de gagner en productivité et en maintenance.

4.1.4 Les autres éléments

4.1.4.1 Bootstrap 3

L'une des problématiques de notre projet est de concevoir un portail qui prend en charge la notion de mobilité, afin de répondre à ce problème nous avons décidé de réaliser un site "responsive", c'est à dire dont l'affichage va s'adapter à la taille du périphérique qui va se connecter au site, on parle alors de "responsive design". CSS3 permet via des medias-queries de conditionner l'affichage en fonction du périphérique, toutefois après s'être renseigné sur les différentes solutions possibles pour réaliser du responsive design, notre choix s'est porté sur un framework CSS qui s'appuie sur CSS3 : Bootstrap 3.

Bootstrap 3 est développé par la société Twitter et se trouve très utilisé dans le monde du Web. Ce framework va nous permettre de gagner du temps dans le développement CSS en proposant des composants prêt à l'emploi afin de réaliser des formulaires, des boutons, des éléments de navigation, etc. Mais il permet aussi et surtout de créer plus rapidement et simplement une application Web responsive, car il inclut un bon nombre de classes CSS permettant de rendre les éléments de l'application compatibles avec tous les types d'écran : smartphone, tablette et écran PC.

Bootstrap 3 a été choisit par rapport à d'autres solutions, car ce dernier dispose d'une grande communauté, d'une bonne maturité et d'un large choix de templates de qualité pour créer un site Web responsive.

4.1.4.2 Doctrine 2

Doctrine 2 est l'ORM le plus populaire pour le langage PHP. Un ORM donne l'illusion de travailler avec un système de gestion de base de données objet plutôt que de travailler avec un système de base de données relationnelles classique. Ceci facilite la vie du développeur, améliore la lisibilité du code (et donc sa maintenance) et permet au développeur de coder plus rapidement en utilisant un ensemble de fonctions prédéfinies par Doctrine 2. Cet ORM fournit également un langage de requête simplifié appelé Doctrine Query Language (DQL), ce langage donne la possibilité au développeur d'utiliser une syntaxe qui permet de réaliser des traitements plus complexes qui ne peuvent être effectuées avec l'aide des fonctions prédéfinies de Doctrine 2. Plutôt que d'utiliser une autre solution d'ORM pour PHP comme Propel, nous avons décidé d'utiliser Doctrine 2 car ce dernier est parfaitement intégré à Symfony 2, de plus l'équipe de développement dispose déjà de compétences sur Doctrine, ce qui réduit le temps d'apprentissage.

4.1.4.3 jQuery

JQuery a déjà été présenté précédemment dans le document, pour rappel il s'agit d'une librairie développée en JavaScript dont le principal objectif est de gagner du temps dans le développement des applications, il s'agit même de son slogan : "write less, do more". Il existe des alternatives à cette librairie, cependant jQuery propose un nombre très important de plugins, dispose dune communauté énorme et de plus est très bien couplée avec le framework Bootstrap, son choix est donc pertinent.

4.1.4.4 Twig

Également développé par Sensio Labs (société qui est à l'origine de SF2), Twig est un moteur de template pour PHP. L'intérêt principal d'avoir recours à un moteur de template dans le développement est de séparer clairement la couche de présentation de l'application. Twig sera finalement un "pseudo langage" qui sera intégré aux pages HTML, ce pseudo langage apportera des fonctions au sein de la page Web afin de gérer facilement le rendu et il permettra également de rendre me contenu de la page dynamique à l'aide d'instructions de contrôle comme des conditions ou des boucles. Le choix de Twig comme moteur de template PHP plutôt qu'un autre (Smarty ou Mustache) est que son intégration est parfaite dans SF2. De plus, Twig est intégré par défaut dans la distribution standard ce qui facilite sa mise en place et son paramétrage. En effet, SF2, Doctrine 2 et Twig sont trois technologies différentes et complètement indépendantes, mais elles sont cependant parfaitement couplées entre elles dans la distribution de SF2.

Il est vrai que toutes les opérations réalisées avec Twig sont aussi réalisables en PHP au sein d'une page HTML, cependant pour être convaincu des avantages de Twig, voici deux exemples qui produisent un résultat identique :

```
1  // Exemple 1 : Affichage d'une variable "échappée"
   <?php echo htmlspecialchars($var, ENT_QUOTES, "UTF-8") ?> //en PHP
3  {{var | e}} //en Twig

5  //Exemple 2 : Affichage de la date au format FR (jj/mm/aaaa) d'un objet métier qui est
       stockée en base de données au format US (aaaa-mm-jj)
   // En PHP
7  <?php
   $dateUs = $article->getDateCreation();
9  $dateFr = new \DateTime($dateUs)->format('d/m/Y'); // Transformation du format pour l'
       affichage
   echo $dateFr;    // Affichage de la date
11  ?>
   // En Twig
13 {{article.dateCreation | date('d/m/Y') }}


15
```

4.1.4.5 ExtJS

ExtJS est un framework MVC dédié au développement d'application internet client riche, il est proposé par la société Sencha qui propose un support et des outils facilitant le développement d'applications à l'aide de ExtJS. Ce framework permet de réaliser un travail de qualité avec des composants graphiques personnalisables (graphiques, tables, calendrier, etc.), un gestionnaire de fenêtres permettant de positionner facilement les composants et des thèmes personnalisables. Les grandes

forces de cette librairie sont la rapidité de développement pour une qualité visuelle élevée, une documentation riche et une communauté d'utilisateurs active et importante. Nous avons choisit ce framework, car il n'existe pas de réel concurrent, en effet les autres alternatives souffraient d'un manque de qualité et de documentation (GWT par exemple) ou n'avaient pas la même couverture fonctionnelle en ne proposant que des éléments graphiques sans proposer un réel cadre de développement (jQueryUI ou KendoUI).

4.1.4.6 GIT

Jusqu'à présent, le code source des précédentes applications était géré avec le gestionnaire de versions Subversion (SVN). Pour des raisons de sécurité, de maintenance et de souplesse nous avons décidé de mettre en place GIT en réimportant l'historique des anciennes applications.

Un grand avantage de GIT par rapport à Subversion est la gestion des "commit" (enregistrement des modifications). Lorsqu'un développeur effectue un commit, ce commit n'existe que pour lui, il peut alors bénéficier des fonctionnalités du logiciel de gestion de versions sans pour autant envoyer les modifications directement au serveur via l'action "push". Dans le cas de Subversion, un commit était systématiquement envoyé au serveur et ce dernier était alors disponible pour l'ensemble des autres développeurs travaillant sur ce projet, cet aspect pouvait poser des problèmes si les modifications étaient erronées ou incomplètes. Cette nouvelle gestion des commit offre ainsi une plus grande flexibilité au développeur et lui permet de bénéficier des avantages du système de gestion de versions sans toutefois altérer le code source du référentiel.

Un autre point intéressant à noter est le système de branches de GIT, ce dernier est souple et particulièrement puissant, ce qui justifie la grande utilisation et popularité de GIT dans le monde du développement. Ces branches permettent de créer des évolutions sur le projet sans impacter toutefois la branche "courante" (nommée master), il est alors possible de pouvoir travailler sur une nouvelle fonctionnalité importante d'un projet tout en continuant la maintenance et l'intégration continue de petits changements sur la branche principale. Cette fonctionnalité est parfaitement implémentée dans GIT.

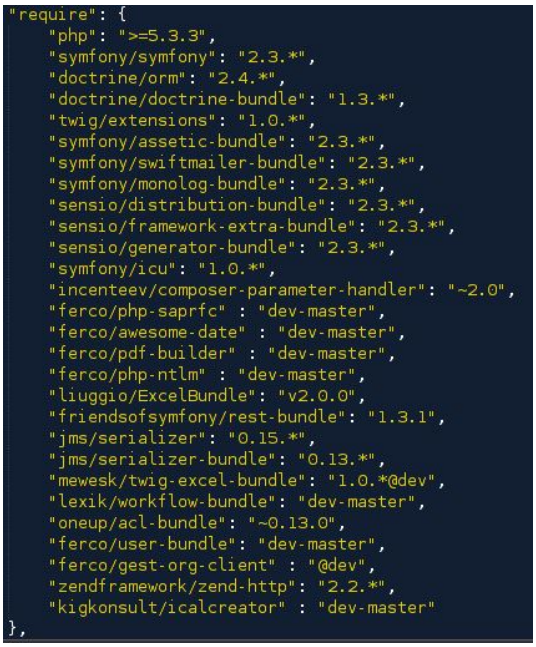
4.1.4.7 Composer

Composer est un outil récent, qui a pour but de gérer les dépendances en PHP. Les dépendances, ce sont toutes les librairies dont le projet dépend pour fonctionner. Composer a pour objectif de gérer toutes les dépendances d'un projet PHP, et ceci permet de résoudre plusieurs problématiques lorsque l'on utilise des librairies externes :

- la mise à jour des librairies : il faut veiller à avoir les dernières versions afin d'obtenir des corrections de sécurité et/ou des évolutions de fonctionnalité,
- les librairies peuvent elles aussi avoir des dépendances vers d'autres bibliothèques et sans outil adéquat ceci va grandement complexifier la maintenance du projet : si une des librairies dépend d'autres librairies, cela nous oblige à gérer l'ensemble de ces dépendances (installation et mise à jour),
- les librairies ont chacune leurs paramètres de configuration (autoload), et il faut gérer leur autoload pour chacune d'entre elles.

Composer est l'outil que propose la communauté de PHP afin de traiter chacun de ces points, l'utilisation de composer implique alors un gain important de temps et de sérénité pour la réalisation d'un projet PHP. L'utilisation de composer se fait à travers un fichier de configuration "compo-

ser.json". La figure 4.2 montre un extrait du fichier de déclaration des dépendances que va utiliser le logiciel.



```
"require": {
    "php": ">=5.3.3",
    "symfony/symfony": "2.3.*",
    "doctrine/orm": "2.4.*",
    "doctrine/doctrine-bundle": "1.3.*",
    "twig/extensions": "1.0.*",
    "symfony/assetic-bundle": "2.3.*",
    "symfony/swiftmailer-bundle": "2.3.*",
    "symfony/monolog-bundle": "2.3.*",
    "sensio/distribution-bundle": "2.3.*",
    "sensio/framework-extra-bundle": "2.3.*",
    "sensio/generator-bundle": "2.3.*",
    "symfony/icu": "1.0.*",
    "incenteev/composer-parameter-handler": "~2.0",
    "ferco/php-saprfc" : "dev-master",
    "ferco/awesome-date" : "dev-master",
    "ferco/pdf-builder" : "dev-master",
    "ferco/php-ntlm" : "dev-master",
    "liuggio/ExcelBundle": "v2.0.0",
    "friendsofsymfony/rest-bundle": "1.3.1",
    "jms/serializer": "0.15.*",
    "jms/serializer-bundle": "0.13.*",
    "mewesk/twig-excel-bundle": "1.0.*@dev",
    "lexik/workflow-bundle": "dev-master",
    "oneup/acl-bundle": "~0.13.0",
    "ferco/user-bundle": "dev-master",
    "ferco/gest-org-client" : "@dev",
    "zendframework/zend-http": "2.2.*",
    "kigkonsult/icalcreator" : "dev-master"
},
```

FIGURE 4.2 – Extrait du fichier de configuration des dépendances pour Composer

Composer offre un ensemble complet de directives qui sont utilisables dans ce fichier de configuration, la liste complète est disponible sur le site Internet de l'outil [Composer].

Après le lancement d'une commande "composer install", le logiciel va rechercher les versions des librairies compatibles avec les contraintes indiqués dans le fichier de configuration, puis va lancer le téléchargement et l'installation de ces librairies.

4.1.4.8 Bower

Bower est un utilitaire de gestion des dépendances qui a pour but de gérer les librairies front-end que peuvent utiliser des projets Web. Cet outil est basé sur GIT et Node.js (environnement d'exécution JavaScript, qui permet notamment la développement d'application JavaScript côté serveur). Tout comme le logiciel Composer pour la gestion des codes sources PHP, Bower permet de configurer les différents éléments requis par le projet via un simple fichier de configuration "bower.json" qui doit être déposé à la racine de celui-ci. La figure 4.3 montre un extrait de la partie du fichier qui traite des dépendances du projet.

FIGURE 4.3 – Extrait du fichier de configuration des dépendances pour bower

Un simple commande "bower install" ou "bower update" va exécuter l'installation ou la mise à jour des librairies spécifiées par le fichier de dépendances. Si des conflits de version arrivent, alors bower va demander à l'utilisateur d'arbitrer les décisions à prendre, c'est à dire de prendre la décision de la version qui doit être installée.

4.2 Conception du projet

Lors de la phase d'étude du projet, nous avions décidé de séparer certaines parties du nouveau PAE dans des applications autonomes qui exposeront leurs fonctionnalités à travers des services Web. Ce choix a été justifié par plusieurs raisons, tout d'abord le retour d'expérience du portail Intranet a permis de mettre en évidence un point important : la base de données utilisateurs et la gestion des droits étaient une partie importante et complexe (notamment la synchronisation des informations avec les systèmes externes) du code source de l'application, mais cette partie n'apportait que peu de valeur ajoutée au niveau métier dans les applications hébergées dans l'Intranet. De plus, certaines applications en dehors du portail Intranet avaient besoin de ces informations pour fonctionner, c'est pourquoi mettre en place et rendre disponible dans le SI de l'entreprise une application dédiée à la gestion des utilisateurs et de leurs droits est un choix pertinent qui va permettre de simplifier la maintenance. Le second point est comme annoncé dans le cahier des charges, la génération des *messages de logs* sur les objets métiers dans le but d'améliorer la sécurité et la traçabilité des modifications apportées. Afin de répondre à ce besoin, nous avons imaginé un système permettant de recueillir les *messages de logs* métiers d'un système quelconque et de les centraliser au sein d'une base de données. En dehors de ces applications autonomes, durant la phase d'étude nous avions observé qu'il était possible de développer en PHP des composants indépendants via le système de gestion de dépendances Composer, nous avions donc également décider d'utiliser ce principe afin de réaliser des composants PHP que nous avions jugé utile de rendre distribuables et autonomes dans le but de les réutiliser dans différentes applications.

4.2.1 Architecture générale

Dans le cadre de notre projet qui est de réaliser un nouveau PAE, nous aurons plusieurs développements à effectuer :

— Les composants PHP techniques distribuables qui seront disponibles sur l'ensemble du système d'information.
— Le portail applicatif (*portail App*), qui sera l'application Web dont le rôle sera d'héberger les futures applications métiers développées pour l'entreprise. Cette dernière devra utiliser le fra-

mework Symfony 2 et mettra en place un socle technique (paramétrage et surcouche des framework utilisés) afin d'obtenir un développement rapide des applications métiers.
— L'application pilote "la gestion des déplacements", qui sera une application métier au sein du *portail App*.
— L'application gestion de l'organisation (*gestOrg*), qui sera l'application dédiée à la gestion des salariés, de la diffusion de leurs données au sein du système d'informations. Il est à noter que la *gestOrg* sera interfacée avec le progiciel SAP.
— L'application d'historisation des événements (*ObjectsLogger*), qui va permettre la centralisation des *messages de logs* et qui sera interfacée avec le *portail App*.

Le développement du PAE se trouve être l'ensemble de ses éléments combinés, cependant comme le *portail App* est l'application Web frontale que la personne utilisera et qui va héberger les applications métiers, traiter les requêtes des clients et orchestrer tous les éléments développés, il peut être considéré par abus de langage comme le PAE. C'est pourquoi quand nous citerons le *portail App*, nous évoquerons finalement le PAE avec ses interfaces.

Cette approche de vouloir créer des applications et des composants dédiés à un service précis aura pour but de réaliser une application avec une architecture n-tiers. Ce développement va permettre de simplifier la maintenance et d'ouvrir des fonctionnalités en tant que "services" sur l'ensemble du système d'informations. Il s'agit là du premier pas de la société FERCO dans une démarche d'architecture orienté services (SOA).

La figure 4.4 illustre l'architecture du PAE en présentant ses applications et composants.

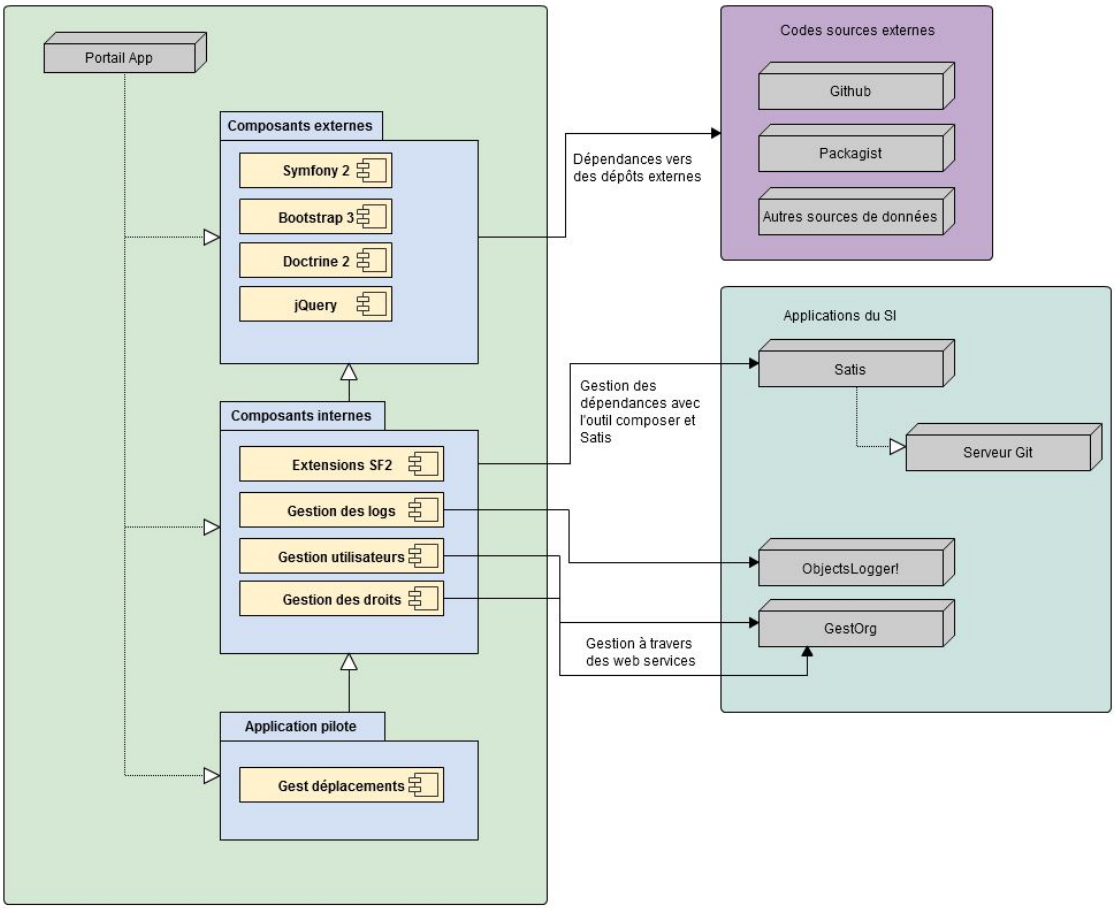


FIGURE 4.4 – Architecture générale du PAE

Description de l'architecture physique

La mise en place de ce projet nécessite un ensemble de serveurs et d'environnements qu'il est nécessaire de faire communiquer. La figure 4.5 illustre la structure physique globale du projet à travers un diagramme de déploiement. Afin de simplifier la lisibilité, uniquement les liens d'API de la *gestOrg* et de *ObjectsLogger* ont été représentés (il existe des liens entre la *gestOrg* et SAP par exemple).

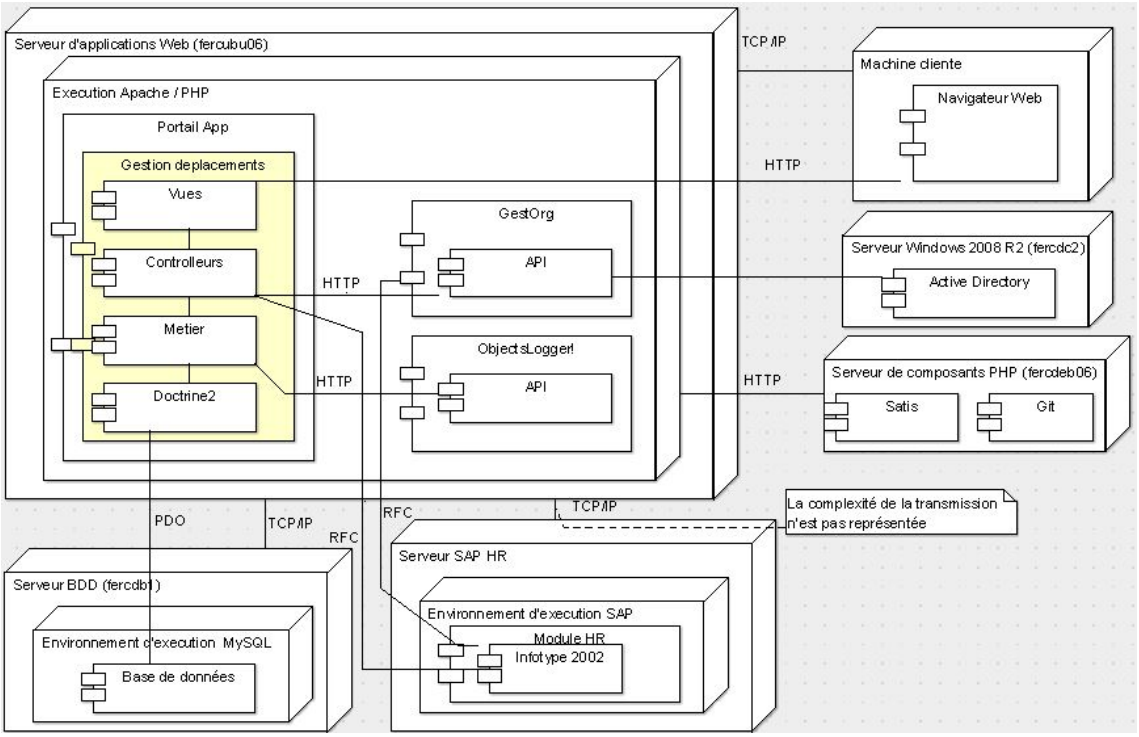


FIGURE 4.5 – Diagramme de déploiement

4.2.2 Architecture détaillée

4.2.2.1 Le *portail App*

Le *portail App* (le PAE) est une application MVC qui va utiliser principalement les framework Symfony 2, Bootsrap 3, jQuery et Doctrine 2. Comme présenté sur le schéma de l'architecture du projet, le *portail App* utilisera :

- les composants externes : ce sont les librairies créées par des communautés disponibles sur le Web. Ces librairies sont importées et configurées pour le bon fonctionnement du portail d'applications. Le logiciel composer va prendre en charge leurs gestions et leurs dépendances.
- les composants internes : ce sont des composants développés par la société FERCO et qui répondent à un besoin technique précis. Ces composants peuvent être distribuables sur l'ensemble du système d'informations via le logiciel composer ou peuvent être développés directement dans le socle technique du *portail App*. Les composants distribuables sont développés de manière totalement indépendantes et peuvent être utilisés par une application PHP quelconque du système d'informations (par exemple un composant PHP pour la gestion avancée des dates ou encore un composant qui permet de faire appel aux services Web d'une application comme la *gestOrg*).

Comme le montre la figure 4.6, le *portail App* intègre un socle technique qui se compose d'un ensemble d'éléments dépendant des frameworks retenus visant à optimiser le développement des applications Web qui y sont hébergées. Ces éléments traitent des sujets qui touchent toutes les couches de l'application.

Le socle technique et les frameworks associés (principalement Symfony 2) seront le coeur du *portail App* et vont rendre possible le développement rapide des applications métiers.

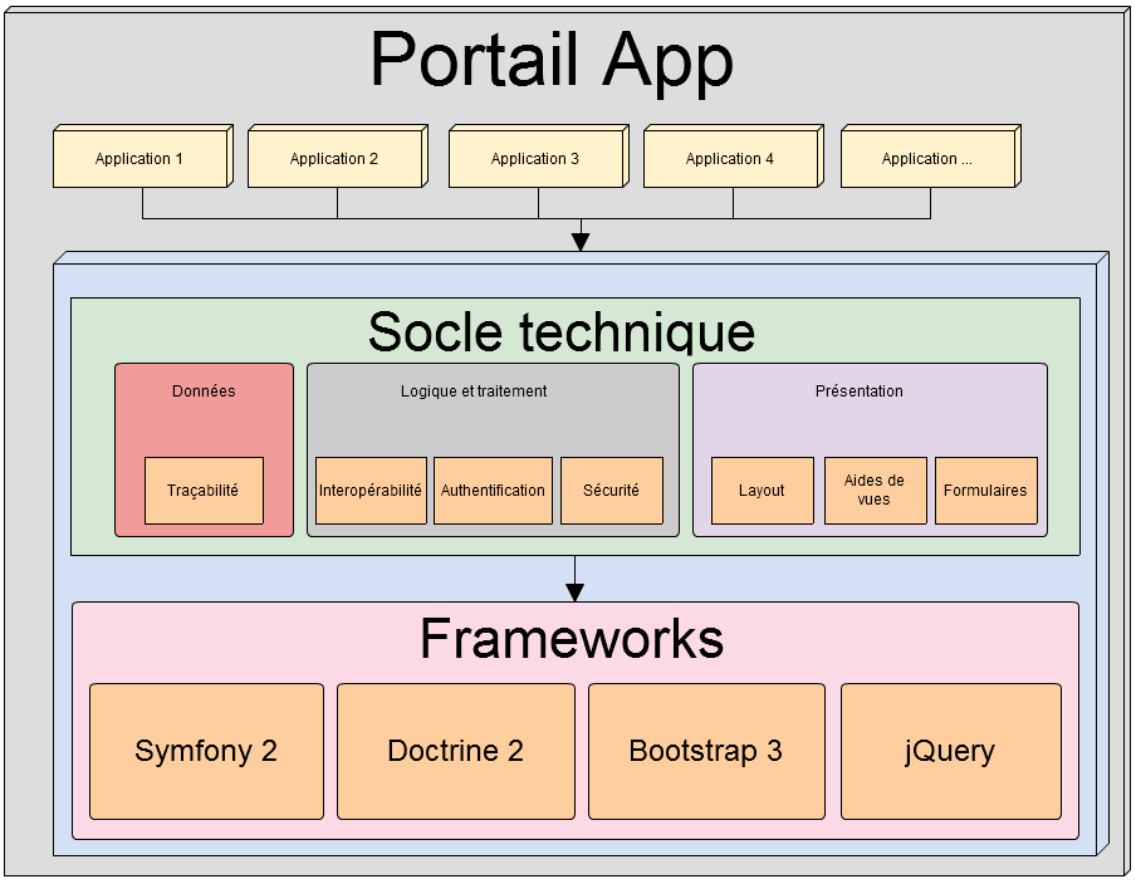


FIGURE 4.6 – Schéma de conception du *portail App*

4.2.2.2 L'application *ObjectsLogger*

ObjectsLogger est l'application dédiée à la centralisation des *messages de logs* applicatifs. Cette dernière a pour objectif de proposer des services Web de type REST qui vont permettre aux applications du SI de stocker les modifications de leurs objets métiers. Cette solution a l'avantage d'être agnostique au langage de programmation utilisé par l'application du SI, tout en permettant à cette dernière de diminuer sa complexité, en effet elle n'a pas besoin de traiter la traçabilité et l'historique des modifications sur les données qu'elle gère. Afin de garder une homogénéité dans le développement de nos applications Web, cette application va utiliser les mêmes framework que le *portail App*.

La figure 4.7 illustre le résultat que l'on souhaite atteindre avec l'application *ObjectsLogger*. Afin de mieux décrire l'objectif attendu, nous avons imaginé connecter une application (fictive) de relation client (CRM) à l'application *ObjectsLogger*. L'application CRM gère une base de données client, et on y retrouve une entité *Client*. Un gestionnaire modifie et enregistre plusieurs fois une instance de la classe *Client*, on peut observer que l'application CRM ne va garder que la dernière information de l'objet au sein de sa base de données, alors que *ObjectsLogger* va garder tout l'historique de l'objet en y associant des informations pour pouvoir identifier l'objet de manière unique (référence de l'application, de sa classe et son identifiant).

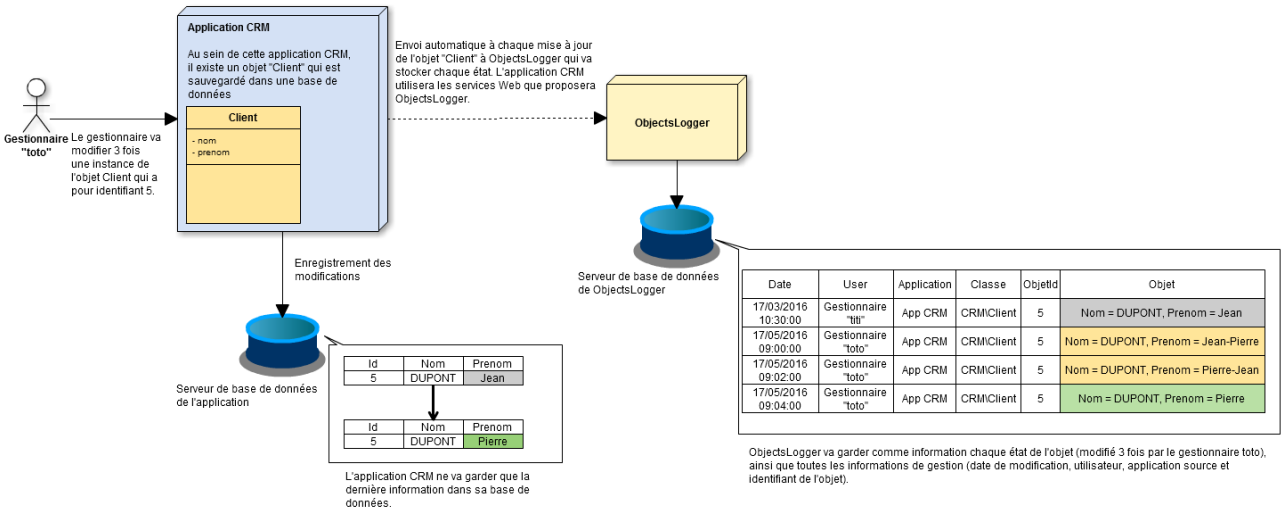


FIGURE 4.7 – Fonctionnement de l'application *ObjectsLogger*

La figure 4.8 illustre le fait que l'application *ObjectsLogger* est agnostique au langage des applications utilisatrices. En effet, ces dernières devront simplement implémenter un écouteur qui va appeler les services Web de l'application *ObjectsLogger*. A chaque modification d'une entité l'application source va alors transmettre les données de l'objet, ainsi que les références de l'application source.

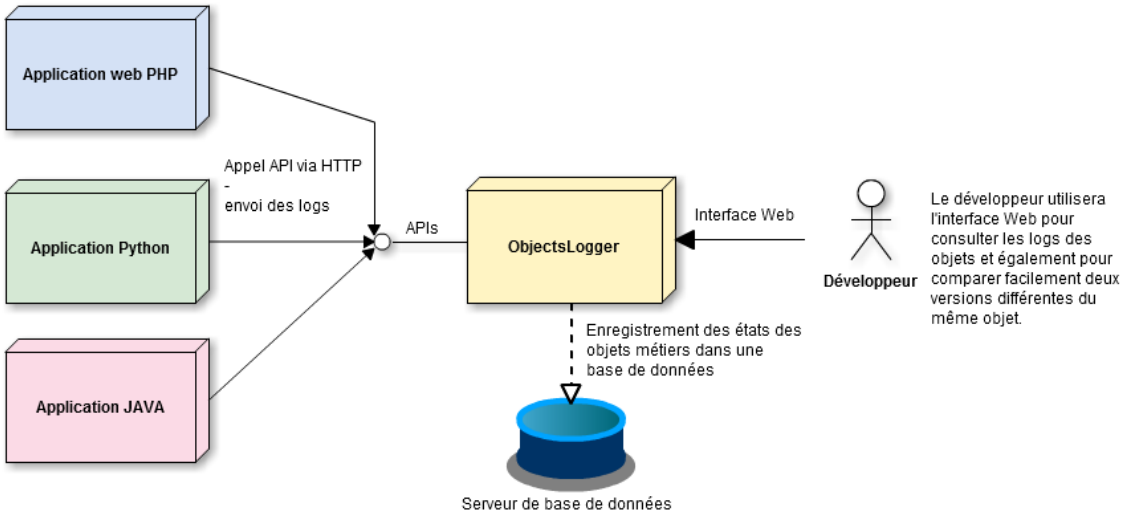


FIGURE 4.8 – Schéma de conception de *ObjectsLogger*

En dehors du simple enregistrement des modifications, on souhaite également avoir la possibilité d'avoir une interface graphique permettant de consulter de manière ergonomique les *messages de logs* enregistrés et disposer d'une manière simple permettant de comparer les différents états de l'objet métier.

Diagramme de classes

La liste des classes nécessaires au développement de l'application peut être illustrée à l'aide d'un diagramme de classes. La figure 4.9 montre les classes à développer dans le cadre de cette application et permet d'avoir une vue globale de la conception.

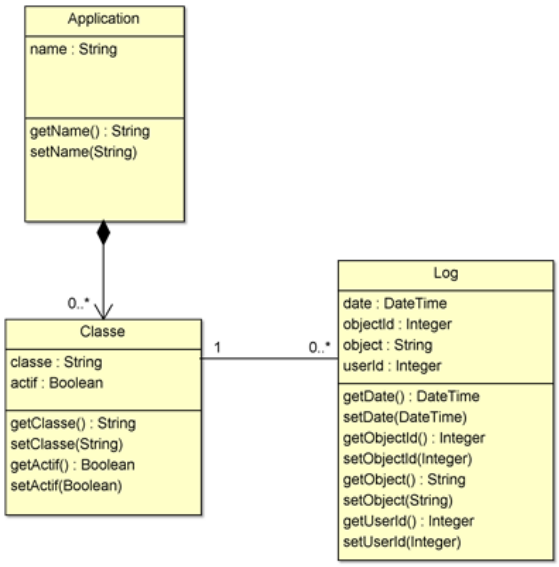


FIGURE 4.9 – Diagramme de classes de l'application *ObjectsLogger*

La classe *Application* va permettre de définir la notion l'application source qui va détenir plusieurs classes. Chaque *Classe* est défini à l'aide de son full qualified class name (FQCN) qui se trouve être l'attribut "classe". Un message de log est donc décrit par la classe *Log* qui est reliée à *Classe*, on retrouve dans chaque *Log* les informations de l'objet, ainsi que les informations de traçabilité (utilisateur, date et classe source).

Afin de mieux décrire ces objets, le diagramme d'objets qui est représenté par la figure 4.10 montre une illustration avec l'exemple précédent de l'application CRM.

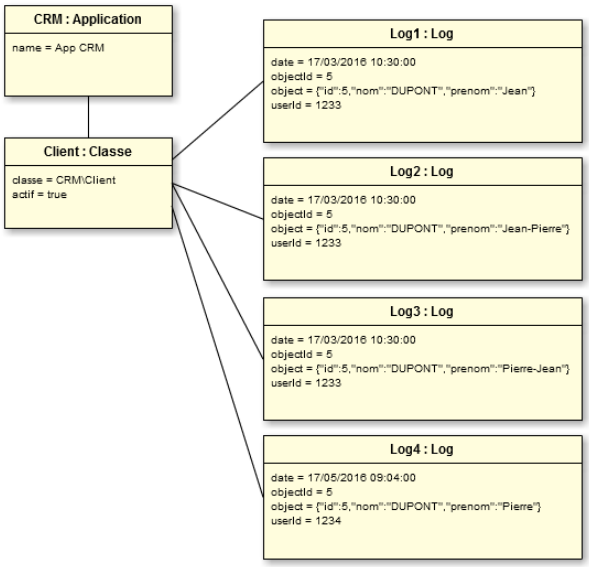


FIGURE 4.10 – Diagramme d'objets de l'application *ObjectsLogger*

Règles de gestion

Suite à l'étude de l'application *ObjectsLogger*, les règles de gestion suivantes ont été exprimées :

1. Une application qui est déclarée dans le système de *ObjectsLogger* possède un nom (name) et contient plusieurs classes.
2. Une classe est identifiée par son FQCN par l'attribut "classe". Le FQCN est la concaténation de l'espace de nom (le namespace) de la classe et son nom. L'attribut "actif" permet d'activer ou désactiver l'enregistrement des *messages de logs* de cette classe.
3. Un message de log est un événement qui enregistre les modifications survenues sur un objet métier (création et modification). Le couple application et FQCN permet de définir un contexte unique pour déterminer sur quelle classe porte la log. Cette dernière enregistre la date de l'événement, l'identifiant de l'objet "objectId" dans la base de données source et une sérialisation (représentation sous forme de chaîne de caractères de l'objet) via l'attribut "object". On retrouve également l'identifiant de la personne qui est originaire du message de log à l'aide de l'attribut "userId".

Modèle physique de données

La figure 4.11 montre le modèle physique de données qui a été utilisé pour générer la base de données de l'application sur notre serveur MySQL.

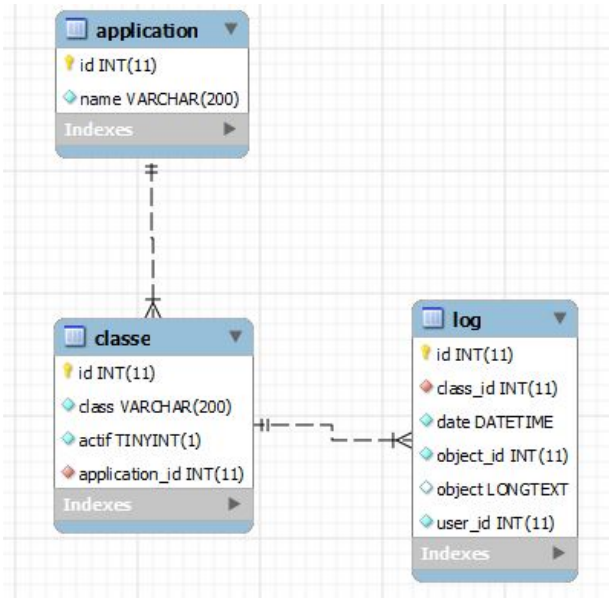


FIGURE 4.11 – Modèle physique de données de l'application *ObjectsLogger*

Diagramme de séquence

L'enregistrement des événements à l'aide de l'application *ObjectsLogger* s'effectue à travers l'appel de deux services Web. La première requête va avoir pour but de récupérer l'identifiant de la classe à traiter depuis la base de données en fonction de l'application source et du FQCN de la classe de l'objet métier à traiter. Une fois cet identifiant récupéré par l'application source, cette dernière doit sérialiser l'objet métier afin de transmettre les informations de l'objet à une seconde requête qui va réaliser l'enregistrement dans la base de données. Ce scénario est illustré avec la figure 4.12 à l'aide d'un diagramme de séquence.

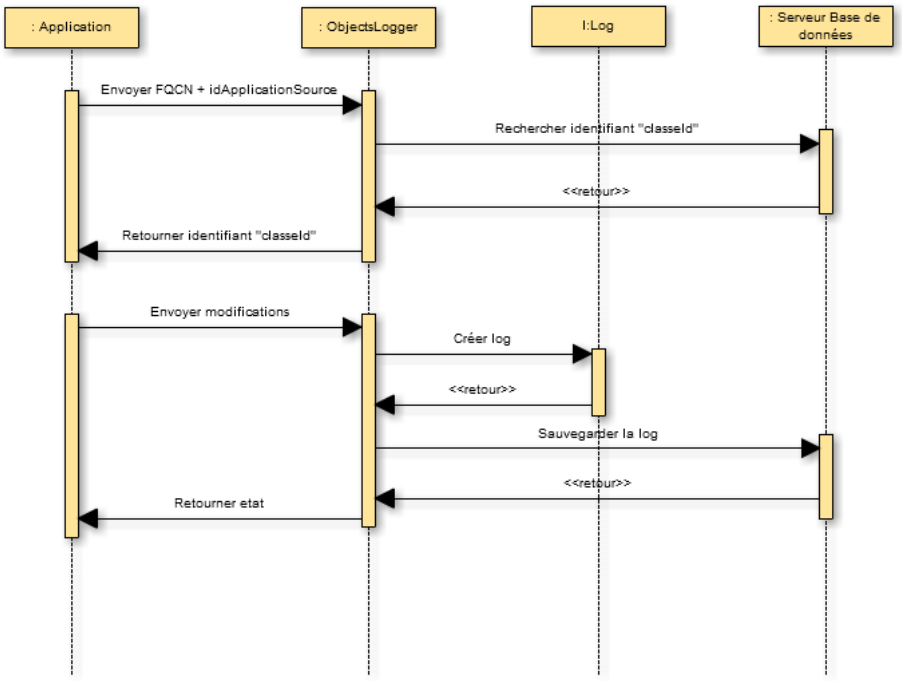


FIGURE 4.12 – Scénario de l'enregistrement d'un *message de log*

4.2.2.3 L'application *gestOrg*

Pour rappel, la *gestOrg* est l'application qui va rendre disponible dans l'ensemble du système d'informations les données des salariés de FERCO ainsi que leurs rôles et autorisations informatiques. Ce système permettra notamment au PAE d'accéder à l'ensemble des informations des utilisateurs via des API (services Web de type REST). Pour des raisons de sécurité et de confidentialité, nous allons distinguer deux catégories d'API : publiques et privées. Les API publiques seront accessibles par l'ensemble des postes de l'entreprise de FERCO sans couche de sécurité particulière, alors que les API privées seront protégées par un système de sécurité.

Les données proposées par l'application *gestOrg* sont issues de deux systèmes externes : l'Active Directory (données utilisateurs) et SAP-HR (données salariés). Ces données seront synchronisées quotidiennement par un traitement automatique afin que les services Web fournissent des données à jour. Des données complémentaires (droits, rôles et définition de structure pour regrouper le personnel par services) seront saisies par un gestionnaire.

L'objectif de cette application est donc la gestion du personnel de l'entreprise, en centralisant les données des différentes bases de données. En d'autres mots, son but est de délivrer une application au sein de FERCO, qui va permettre de centraliser la gestion de l'organisation dans le sens « applicatif et validation » et ainsi devenir la référence des données utilisateurs et de fournir ses données à l'aide de services Web aux applications tierces (dont notamment le PAE) du système d'informations.

La figure 4.13 illustre ce point.

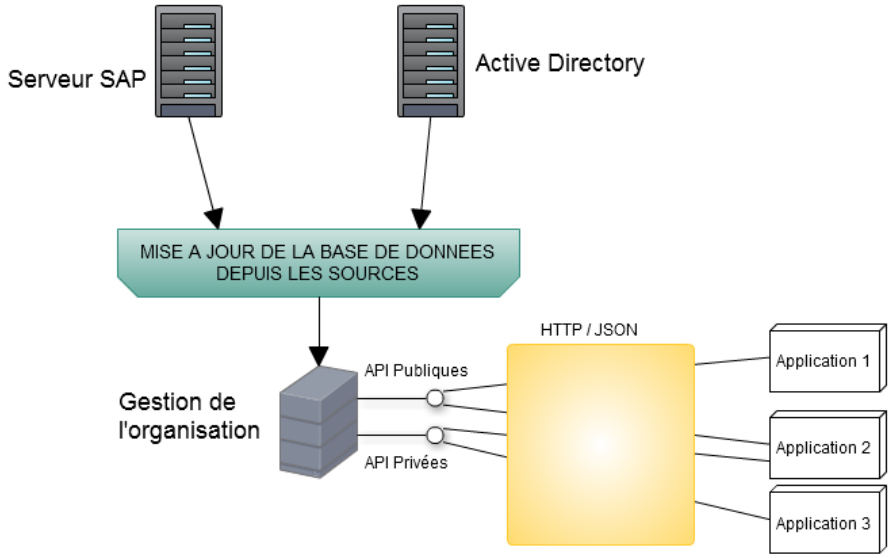


FIGURE 4.13 – Schéma de conception de la gestion de l'organisation

L'application doit permettre de répondre à l'ensemble des questions liées à la structure de l'entreprise. Voici quelques exemples de question qui trouveront une réponse dans cette application : dans quel groupe se trouve le salarié ? Qui est son manager ? Qui dispose des droits de gestion sur ce groupe de personnes ?

Cette application est décomposée en deux parties, tout d'abord le backend qui va s'occuper de l'ensemble de la couche d'accès aux données et de mises à jour, tout en proposant les services Web REST en HTTP/JSON. Ensuite on retrouve le frontend qui est la partie dédiée au gestionnaire et qui permet d'interagir avec les données de l'application. Afin de proposer une meilleure ergonomie et expérience utilisateur au gestionnaire, nous avons choisi d'utiliser ExtJS pour le développement du front-end de cette application, le back-end sera lui développé avec l'aide de Symfony 2 et Doctrine 2.

Définitions

Dans cette application, nous avons utilisé des notions qu'il est nécessaire de définir pour une meilleure compréhension du fonctionnement de l'application.

- Personne : elle peut être un salarié de l'entreprise, un externe, un intérimaire ou encore un stagiaire. Une personne dispose d'informations générales (date de naissance, nom, prénom, etc.) et peut aussi disposer d'un compte utilisateur.
- Compte utilisateur : il s'agit des informations du compte informatique qui est utilisé par une personne.
- Groupe : aussi appelé service, représente les différentes unités structurelles qui composent l'entreprise (service informatique, service achats, etc.).
- Rôle applicatif : rôle sur une application (une personne peut utiliser l'application X ou Y).
- Rôle de gestion : rôle qu'une personne peut avoir sur un service (exemple : rôle de responsable - le responsable informatique peut valider les congés du personnel qui se trouve dans le groupe "Informatique").
- Autorisation applicative : autorisation sur un rôle applicatif. Une autorisation possède une date de début et une date d'expiration.

— Autorisation de gestion : autorisation sur un rôle de gestion. Une autorisation possède une date de début et une date de fin.

L'application *gestOrg* doit fournir deux principaux ensembles de fonctionnalités, afin de remplir ses objectifs fonctionnels : une interface graphique ergonomique pour la manipulation des données et un ensemble de services Web pour ouvrir ses fonctionnalités à des applications tierces.

L'interface de gestion

L'interface graphique doit proposer une gestion simple et efficace de la totalité des informations du personnel interne ou externe à Ferco. L'administrateur de l'application doit pouvoir réaliser un certain nombre de manipulations.

En ce qui concerne la gestion des personnes, il doit pouvoir :

1. créer une nouvelle personne,
2. modifier les informations d'un utilisateur,
3. rechercher un ensemble d'utilisateurs suivant de nombreux critères de recherche (nom, prénom, matricule, type (externe, stagiaire, interne), service, fonction),
4. affecter un compte utilisateur à une personne,
5. retirer le compte utilisateur de la personne,
6. ajouter/retirer des autorisations applicatives,
7. ajouter/retirer des autorisations de gestion.

Au niveau de la gestion des groupes, l'administrateur doit avoir la possibilité de :

1. consulter l'arbre de hiérarchie des groupes (exemple : tous les services sont sous le groupe "direction",
2. modifier les informations d'un groupe,
3. créer un groupe,
4. voir les groupes inactifs (les anciens groupes, par exemple "peinture").

Concernant la gestion des rôles, il doit pouvoir :

1. créer un rôle (applicatif, ou de gestion),
2. modifier les informations d'un rôle (son nom),
3. supprimer un rôle.

Pour la gestion des comptes utilisateurs, l'administrateur pourra :

1. afficher l'ensemble des comptes utilisateurs,
2. effectuer une recherche en fonction du login.

De plus, l'application doit fournir une interface qui permet de naviguer rapidement entre les différentes fonctionnalités. Elle doit aussi fournir une sécurité optimale, car la consultation ou la modification de ce type de données est sensible (taux d'activité d'une personne, son statut de salarié, etc.).

Enfin, il doit être possible d'effectuer une synchronisation des bases de données. Cette synchronisation correspond à la centralisation des informations des différentes sources de données externes :

SAP-HR et l'Active Directory. En effet, l'ensemble des informations concernant les utilisateurs se trouvent sur l'Active Directory et les données salariés sont elles disponibles dans le système SAP-HR. Il faut donc récupérer ces informations à l'aide d'API délivrées par ces deux systèmes externes, puis les regrouper sur la base de données centralisée de l'application. Même si un traitement automatique est lancé chaque jour, le gestionnaire garde la possibilité de relancer cette synchronisation depuis l'interface graphique (en cas d'éventuels problèmes techniques).

Les services Web

La seconde partie de l'application est de fournir un ensemble de services Web permettant de consulter les informations présentes sur la base de données centralisée. Suivant des adresses bien distinctes, on doit pouvoir accéder à l'ensemble des informations de l'application, au format JSON.

L'utilisateur final de l'application dispose d'une documentation complète décrivant chaque URL et leur type/format de paramètres. Par exemple, l'URL http ://adresse_de_l_application/personnes/find ?id=2510 doit renvoyer, au format JSON, l'ensemble des informations de la personne ayant un matricule égal à 2510. Le retour JSON ci-dessous illustre un exemple simplifié (toutes les données ne sont pas présentes) de l'appel de cet API.

```
[
  {
  "id": 2510,
  "nom": "DILLENSCHNEIDER",
  "prenom": "Vincent",
  "actif": 1,
  "phone": "3178",
  "taux_activite": 100,
  "statut_sal": "CA",
  "fonction": "Chef de projet",
  "code_gest": "10",
  "sexe": "H",
  "groupe":
  {
    "id":110000,
    "nom": "Informatique",
    "actif": 1
  },
  "compte":
  {
    "id":  38,
    "email": "v.dillenschneider@ferco.fr",
    "actif":  1,
    "login": "dillenschneider_v",
    "connexion_internet": 1,
    "pin_photocopieur": 2151
  }
}
]
```

Afin de répondre à l'ensemble de nos besoins, l'application doit fournir les services suivants :

- renvoyer les informations d'une ou plusieurs personnes,
- rechercher des personnes en fonction de critères de recherche, par exemple tous les cadres actifs de la société,
- rechercher la liste des collaborateurs d'un manager (c'est-à-dire l'ensemble des personnes qui se trouvent sous les groupes qu'une personne gère),

— rechercher les responsables hiérarchiques d'une personne (l'ensemble des responsables qui se trouvent au dessus de la personne en se basant sur la structure de l'entreprise),
— retrouver la liste des groupes sous la responsabilité d'une personne,
— renvoyer la liste des rôles d'une personne,
— retourner les informations d'un ou plusieurs comptes utilisateurs,
— rechercher les comptes utilisateurs suivant différents critères de recherche,
— retourner les informations d'un ou plusieurs groupes,
— rechercher les personnes ayant les autorisations de gestion suivant différents critères de recherche (par rapport à un utilisateur, à un groupe ou un rôle demandé).

Diagramme de classes

La liste des classes nécessaires au développement de l'application peut être illustrée à l'aide d'un diagramme de classes. La figure 4.14 montre le diagramme de classes de l'application *gestOrg*. Pour des raisons de lisibilité, tous les getters et setters n'ont pas été indiqués sur le diagramme.

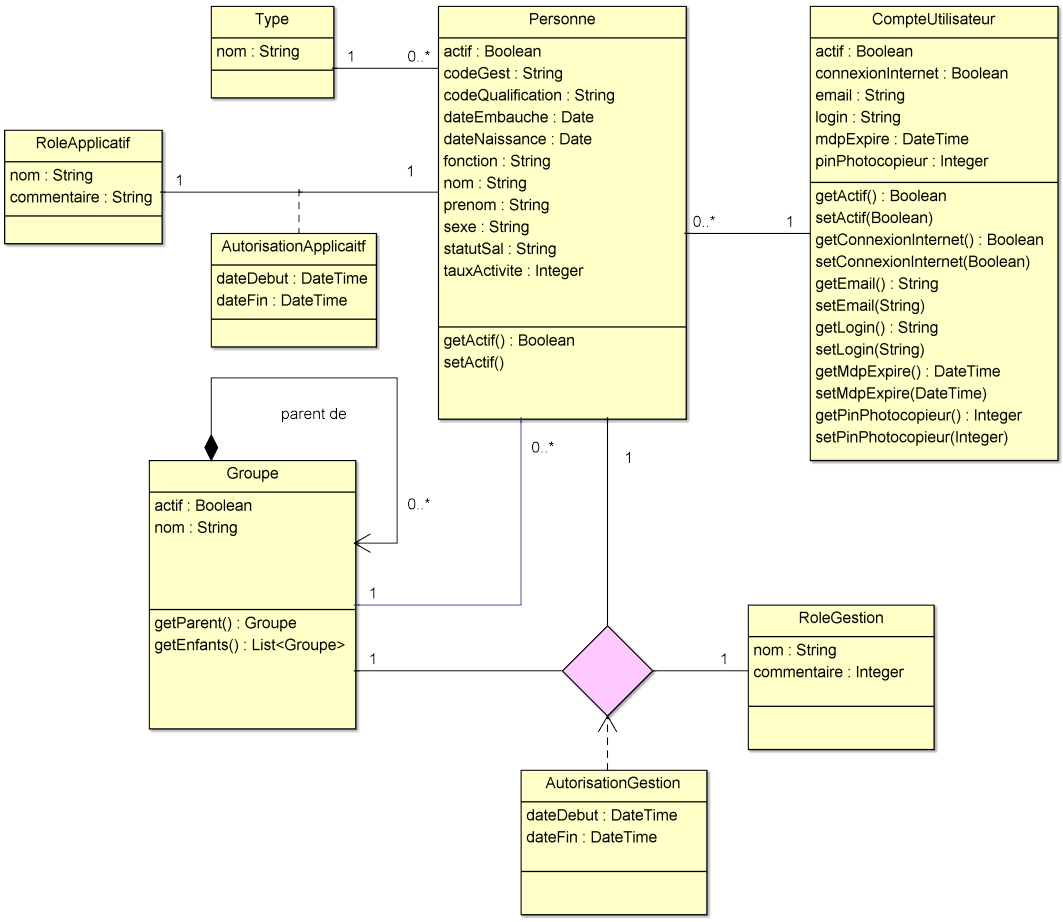


FIGURE 4.14 – Diagramme de classes de l'application *gestOrg*

Modèle physique de données

La figure 4.15 montre le modèle physique de données qui a été utilisé pour générer la base de données de l'application *gestOrg* sur notre serveur MySQL.

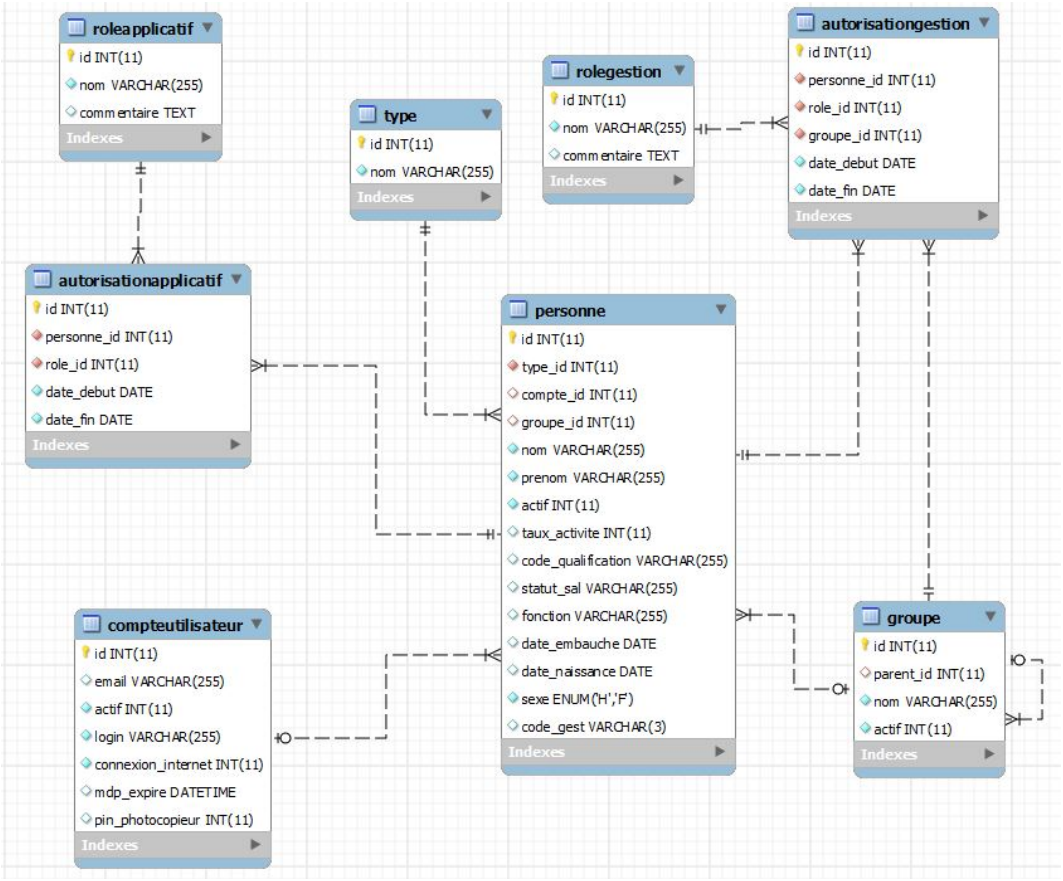


FIGURE 4.15 – Modèle physique de données de l'application *gestOrg*

Règles de gestion

Suite à l'étude de l'application *gestOrg*, les règles de gestion suivantes ont été exprimées :

1. Une personne peut être salarié, stagiaire ou externe à l'entreprise. Elle ne peut appartenir qu'a un seul et unique groupe (même les personnes externes à l'entreprise sont imputées à un groupe de l'entreprise).
2. La personne peut posséder plusieurs autorisations de gestion sur un ou plusieurs groupes et elle peut également disposer de plusieurs autorisations applicatives.
3. Une personne peut disposer d'un compte utilisateur. Ce dernier centralise les informations informatiques de la personne.
4. Un groupe est constitué de personnes et peut être parent d'autres groupes.
5. Les rôles de gestion sont liés à des groupes de personnes contrairement aux rôles applicatifs.
6. Les autorisations possèdent une date de début et une date de fin.

4.2.2.4 Application pilote - gestion des déplacements

La gestion des déplacements est le sujet pilote qui va permettre de développer et tester l'ensemble développé (*gestOrg*, *ObjectsLogger* et le socle technique du PAE).

L'expression des besoins de l'application pilote a été défini précédemment dans ce mémoire (partie 3.3.1.1), le contexte ainsi que les besoins ne seront pas présentés à nouveau dans cette partie du mémoire.

Comme cette application sera intégrée dans le PAE, elle utilise les mêmes technologies et framework à savoir principalement Symfony 2, Bootstrap 3, Doctrine 2 et jQuery.

Diagramme de classes

La liste des classes nécessaires au développement de l'application peut être illustrée à l'aide d'un diagramme de classes. La figure 4.16 montre le diagramme de classes de l'application de la gestion des déplacements. Pour des raisons de lisibilité, les getters et setters n'ont pas été indiqués sur le diagramme.

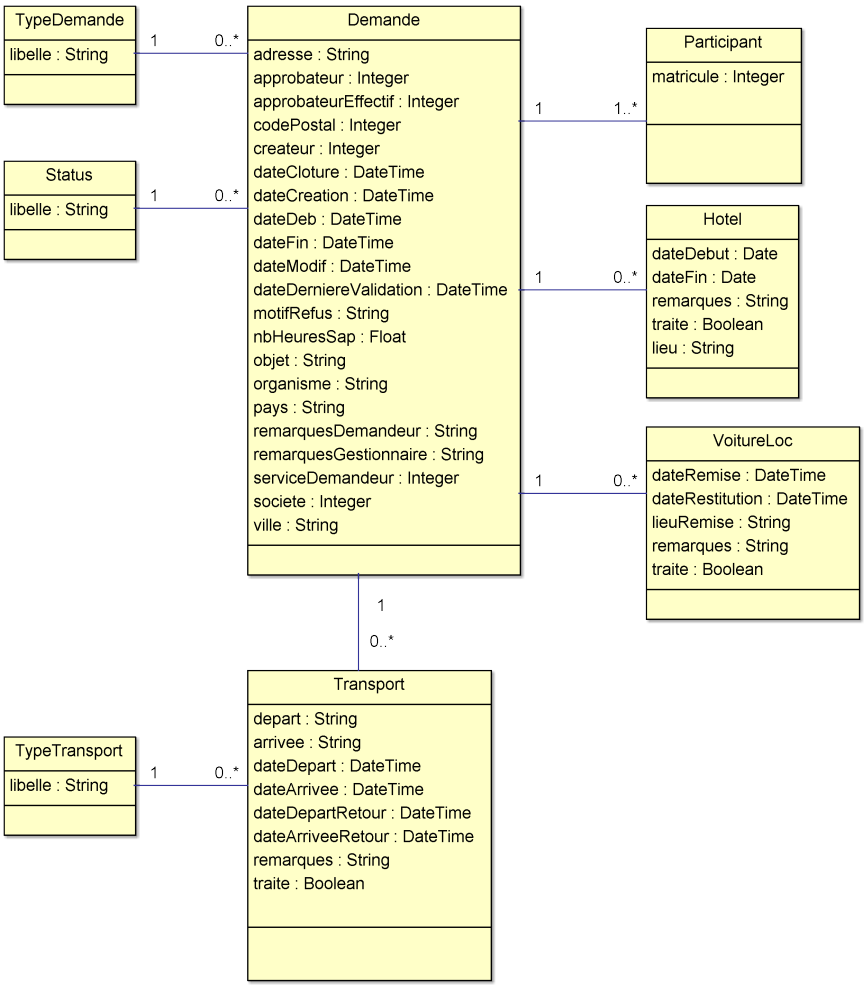


FIGURE 4.16 – Diagramme de classes de l'application gestion des déplacements

Modèle physique de données

La figure 4.17 montre le modèle physique de données qui a été utilisé pour générer la base de données de l'application de gestion des déplacements sur notre serveur MySQL.

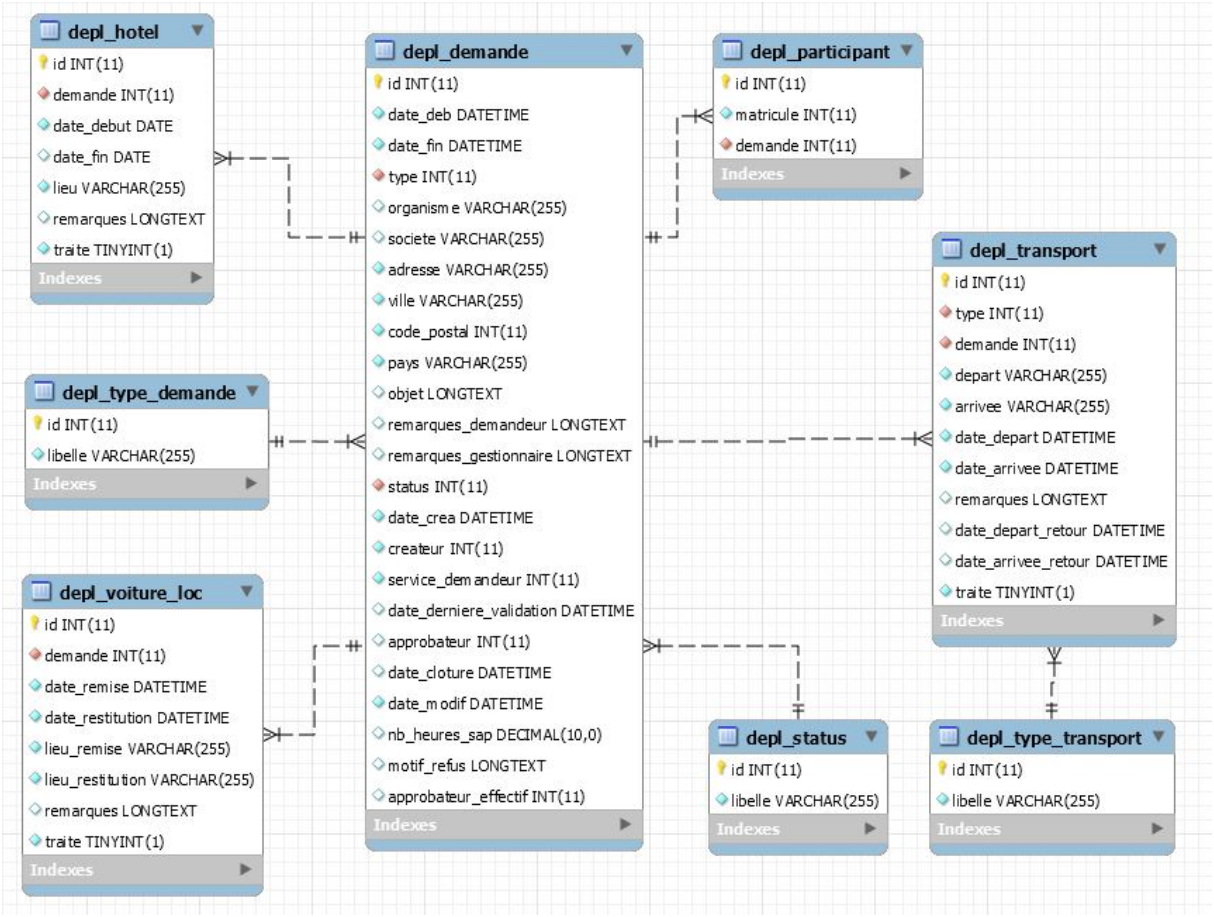


FIGURE 4.17 – Modèle physique de données de l'application de gestion des déplacements

4.3 Installation et configuration

4.3.1 Serveurs

Dans le cadre de notre projet, nous allons utiliser plusieurs serveurs durant la phase de réalisation.

fercubu06

Il s'agit du serveur Web productif qui va accueillir nos applications Web. Il est configuré pour utiliser le logiciel ZendServer CE, ce dernier embarque un serveur Apache, PHP et MySQL (dans notre cas le serveur MySQL n'est pas utilisé car nous avons à la place un serveur dédié). ZendServer CE fournit également un environnement de configuration simple pour gérer Apache et de PHP. ZendServer CE a été installé avec une extension nommée "saprfc" qui offre à PHP un support afin de pouvoir exécuter des fonctions RFC disponibles sur les serveurs SAP.

fercubu06t

Ce serveur est la réplique exacte de l'environnement productif fercubu06, il est utilisé à des fins de tests.

fercdeb06

Le serveur fercdeb06 va être utilisé pour gérer notre système de gestion de versions. Sur cet environnement, les logiciels GIT, Satis et Gitlist (interface Web pour consulter les dépôts Git) sont installés.

fercdb1

fercdb1 est le serveur de base de données productif qui est utilisé par l'ensemble des applications Web développées par la société FERCO. Il va également héberger les nouvelles bases de données nécessaires au projet.

fercdb1-test

Ce serveur est la réplique exacte de l'environnement productif fercdb1, il est utilisé à des fins de tests.

4.3.2 Poste de travail

Depuis la mise à jour technique importante qui a eu lieu en 2010 avec la mise en place du Zend Framework sur l'Intranet, l'environnement de travail du développeur à FERCO n'avait que peu évolué. Le projet d'élaborer un nouveau portail applicatif est aussi le moment idéal pour adapter le poste de travail du développeur afin d'augmenter sa productivité.

Avant notre mise à jour des outils utilisés par le développeur, ce dernier programmait en PHP uniquement avec l'IDE Zend Studio (un dérivé payant d'Eclipse pour PHP) sous Windows 7 et le logiciel Xampp pour faire office de serveur Web local. Ce cadre de travail n'était que partiellement satisfaisant car les serveurs étant sous Linux, il y avait toujours des différences entre la configuration les postes locaux et les serveurs, on pouvait alors passer beaucoup de temps à installer une version spécifique d'un outil ou d'une librairie (quand celle ci était disponible pour Linux et Windows). Cependant le système d'exploitation de nos postes de travail était imposé par l'infrastructure de notre entreprise, car nous devions pouvoir communiquer avec le reste de l'entreprise sans difficultés (intégration dans l'Active Directory, utilisation d'Outlook et de la suite Office, logiciels spécifiques disponibles que sur Windows, etc.). En plus, garder un environnement identique (Windows 7 et Internet Explorer) à celui des autres salariés pour tester nos applications dans les mêmes conditions est un choix cohérent.

Après une discussion sur ce thème entre tous les acteurs (chef de projet, développeur et équipe technique), nous sommes arrivés à une nouvelle configuration de poste qui est optimisée dans le cadre du développement Web à FERCO. Aujourd'hui, nous avons toujours notre poste sous Windows 7 qui est identique à celle de tous les salariés de l'entreprise, mais en plus nous avons à l'aide du logiciel VirtualBox, une machine virtuelle qui fonctionne sous Linux et qui est dédiée au développement Web.

Cette solution de virtualisation offre des avantages : nous avons crée un modèle de machine virtuelle "standard" qui contient toutes les applications par défaut qui sont nécessaires au développement et nos machines virtuelles sont des dérivées de ce modèle, cela permettra notamment de pou-

voir recréer facilement un nouvel environnement de travail si notre poste devait avoir un problème (panne de disque dur ou de carte mère par exemple) ou si un collaborateur supplémentaire devait arriver.

Afin de faciliter la transition entre nos deux environnements, nous sommes maintenant équipés de bi-écran, ce qui permet de passer rapidement d'un système à l'autre en toute transparence.

Notre machine virtuelle est basée sur la distribution LinuxMint avec l'environnement de bureau LXDE (sélectionné pour son côté léger), elle dispose de deux processeurs à 2GHz, de 4Go de RAM et 100 Go de disque dur. Voici ci-dessous la liste des logiciels installés sur notre machine virtuelle dans le but de réaliser du développement Web :

— Netbeans 8 : Cet IDE a été sélectionné à la place de Zend Studio ou Eclipse à la suite de nos tests. Netbeans dans notre environnement a été plus rapide à l'exécution, souffre de moins de problèmes de stabilité que Zend Studio et offre une intégration à des outils très intéressants pour nos développements dont notamment Composer, Symfony 2 et PHPUnit.
— ZendServer CE : Afin de reproduire le plus fidèlement possible notre environnement de production, nous avons mis en place un serveur Web local similaire à celui de nos serveurs.
— MySQL Workbench : Ce logiciel permet la réalisation de Modèle Physique de Données (MPD) à destination de MySQL.
— FireFox : Pour faciliter le développement de nos applications Web, nous avons choisi le navigateur Firefox qui propose de nombreux outils efficaces de debug et de tests dont le très populaire Firebug.
— Pencil : Un logiciel opensource qui permet la réalisation de maquettes.
— Krusader : Un gestionnaire de fichiers avancé très facile à utiliser et qui permet de gérer facilement les fichiers sur nos serveurs.
— GIT : Le client du gestionnaire de versions GIT.

4.4 Organisation du développement

Après avoir analysé les besoins et réaliser la conception générale du projet, nous avons défini une démarche structurée et élaboré un cadre de développement avec la mise en place de GIT et l'intégration du logiciel composer dans notre processus de développement. Nous avons ensuite ordonnancer les tâches de développement à réaliser.

4.4.1 Mise en place de GIT

Nous avons décidé de structurer nos développements avec GIT en travaillant avec trois branches :

— master : il s'agit de la branche courante de développement du projet. Cette branche contiendra des corrections de maintenance et des petites évolutions du projet (inférieur à deux jours de développement),
— features : la branche features va permettre de réaliser des évolutions fonctionnelles plus conséquentes (supérieur à deux jours et inférieur à quinze jours de développement),
— prod : cette branche est celle qui défini la mise en production des modifications sur nos serveurs. Quand nous avons stabilisé notre branche de développement "master" et que nous souhaitons réaliser une mise en production, un import des modifications est réalisé à l'aide de la fonction "merge". Suite à cet import et des tests de non régression., une commande "push" est réalisée pour envoyer les modifications sur notre référentiel. Une fois cette action effectuée,

une commande "pull" permet d'importer les modifications sur nos serveurs de production.

Enfin, hormis les branches citées précédemment, une nouvelle branche "éphémère" peut être créée dans le cas d'évolutions importantes ou lors de la mise en place d'une nouvelle application dans le *portail App*.

4.4.2 Intégration de Composer

Composer va permettre de gérer les dépendances de nos applications et de développer des composants PHP (c'est à dire du code source qui vise à effectuer un traitement précis). Composer fonctionne par défaut avec un dépôt externe qui se nomme *packagist*. Afin de pouvoir fonctionner en mode "privé" c'est à dire d'avoir la capacité de créer des composants uniquement accessibles dans notre entreprise, il est nécessaire de mettre en place une application nommée "Satis".

Satis

Satis (Simple static Composer repository generator) est un programme PHP qui permet de gérer un dépôt privé à destination de Composer. L'utilisation de Satis est obligatoire si on souhaite développer des composants PHP distribuables privés. Satis doit être connecté au système de gestion de versions afin de pouvoir générer un fichier "packages.json" qui contient l'ensemble des informations pour que le logiciel composer puisse fonctionner, Satis génère également une interface Web qui facilite la visualisation des différents paquets délivrés par Satis.

Fonctionnement de Satis

Pour pouvoir proposer un code source sous forme de paquet distribuable avec des dépendances, il faut en premier lieu que le projet soit géré par un système de gestion de versions (dans notre cas GIT), si ce pré-requis est présent, il faut ensuite inclure un fichier composer.json à la racine du projet PHP. Ce fichier composer.json va définir l'ensemble des informations du paquet. L'exemple ci-dessous montre le paramétrage de notre composant PHP gérant les dates avec des fonctions spécifiques sur les jours ouvrés. Les éléments du fichier de configuration sont très simples à comprendre et vont servir à décrire comment gérer ce paquet pour Composer et Satis.

```
{
  "name": "Ferco/awesome-date", // le nom du paquet qui pourra être utilisé par le logiciel composer
  "type" : "library",
  "description": "Librairie PHP de date pour Ferco",
  "keywords": ["date","time"], // mots-clés pour rechercher des paquets dans packgist ou satis
  "authors": [
    {
    "name": "Vincent DILLENSCHNEIDER",
    "email": "v.dillenschneider@ferco.fr"
    }
  ],
  "require": {    // dans cette section nous avons les dépendances du projet PHP
    "php": ">=5.3.0"    // dans ce cas précis, la seule dépendance est au niveau du langage php (>5.3)
  },
  "autoload": { // permet de préciser comment gérer les sources au script de chargement automatique
    "psr-0": { "Ferco\\Date": "src" }
```

```
17
    }
19 }
```

Au niveau de Satis, il est nécessaire de le configurer pour indiquer quels sont les dépôts GIT qui contiennent des composants distribuables sous PHP. Pour réaliser cette action, nous devons ajouter dans le fichier de configuration de Satis (satis.json), une directive afin d'inclure le paquet PHP dans le système de diffusion pour composer, le code ci-dessous montre un exemple de ce fichier de configuration :

```
1     {
        "name": "Ferco repository", // Nom du repository satis
3       "homepage": "http://fercdeb06.ferco.intra/satis/",  //Lien vers la page d'accueil du
      repository
      "repositories": [
5     { "type": "git", "url": "http://172.16.155.2/git/composer/awesome-date.git" },
      { "type": "git", "url": "http://172.16.155.2/git/composer/php-saprfc.git" },
7     { "type": "git", "url": "http://172.16.155.2/git/composer/pdf-builder.git" },
      { "type": "git", "url": "http://172.16.155.2/git/composer/php-ntlm.git" },
9     { "type": "git", "url": "http://172.16.155.2/git/composer/gest-org-client.git" },
      { "type": "git", "url": "http://172.16.155.2/git/composer/sf2-bundles/user-bundle.git" }
11    ],
        "require-all": true // Indique que l'on souhaite obtenir toutes les versions des
      composants
13    }
```

Composer est un logiciel puissant qui permet une gestion efficace des dépendances des codes sources PHP, mais son intégration dans un environnement professionnel est un peu plus complexe. La figure 4.18 montre comment cette intégration a été effectuée au sein de la société FERCO.

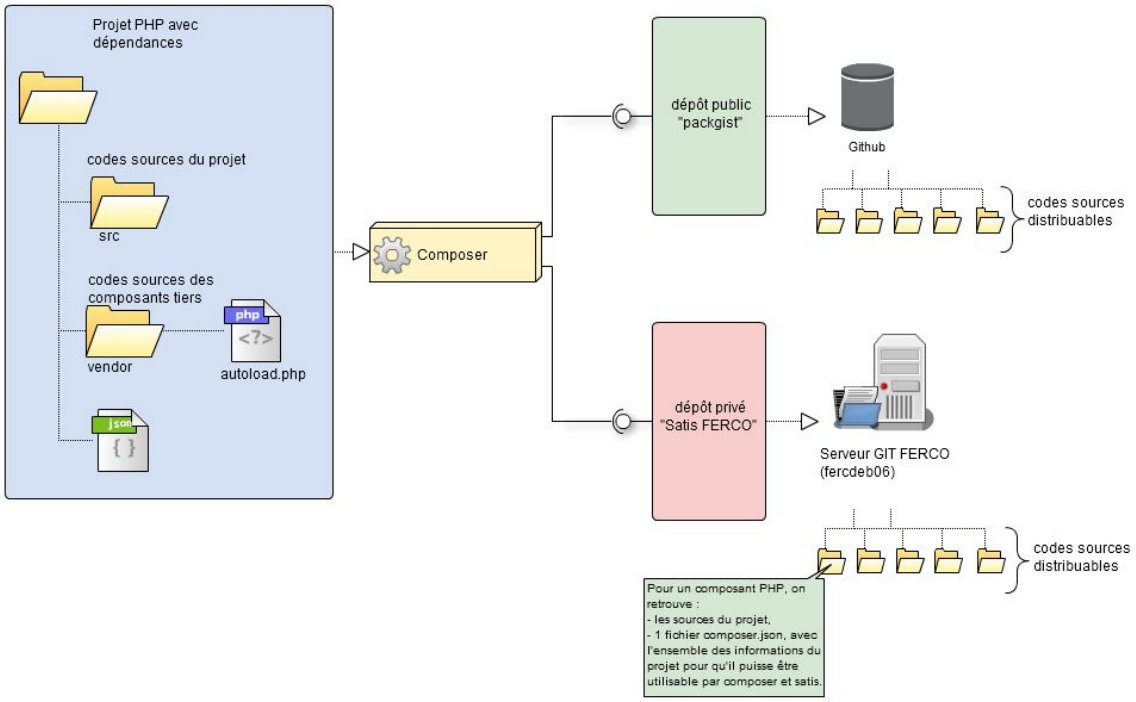


FIGURE 4.18 – Fonctionnement de composer

4.4.3 Itération des tâches de développement

Le développement de l'application est basé sur un cycle de vie itératif et incrémental. Nous avons défini le contenu des itérations de manière à effectuer un travail le plus efficace possible. La figure 4.19 illustre cette approche.

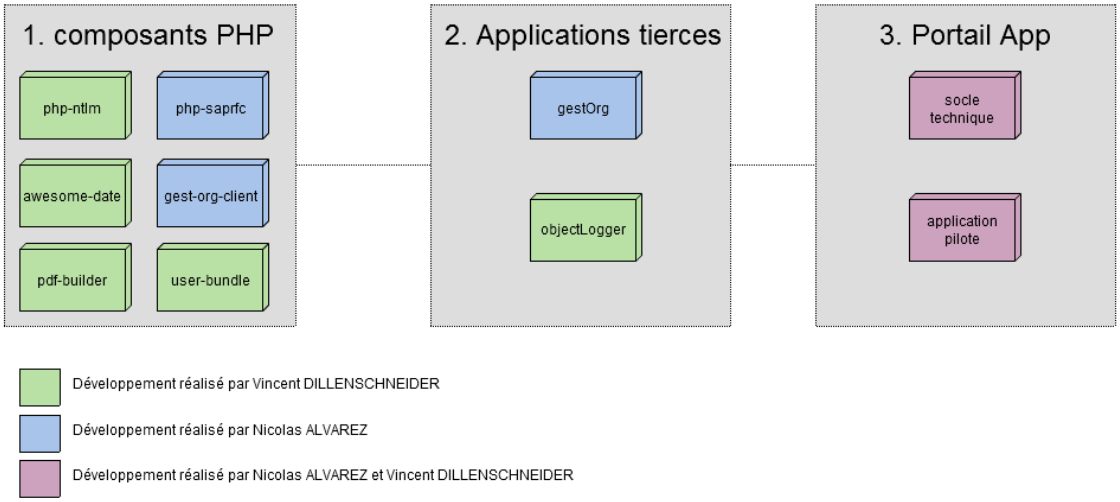


FIGURE 4.19 – Itération des tâches de développement

Comme le montre la figure 4.19, nous avons en premier lieu développer les composants autonomes que nous avions pu identifier lors de la conception. En effet, comme ces composants seront réutilisés dans les autres applications il est donc nécessaire de débuter le développement par ces der-

niers. Une exception cependant sera effectuée pour les composants qui sont reliés à une application tierce (notamment les composants user-bundle et gest-org-client) dont le développement dépend de la réalisation d'une application tierce. Le développement des composants sera conjointement réalisé par Nicolas ALVAREZ et Vincent DILLENSCHNEIDER.

Une fois ce développement terminé nous allons réaliser en parallèle le développement des applications tierces (*gestOrg* et *ObjectsLogger*). Le fait de réaliser ces deux applications de manière simultanée vise à simplifier le processus de développement (pas de conflit à gérer sur le système de gestion de versions) et à responsabiliser chaque développeur sur son sujet. L'application *gestOrg* sera réalisée par Nicolas ALVAREZ et l'application *ObjectsLogger* sera sous la responsabilité de Vincent DILLENSCHNEIDER. Comme la charge de développement est moins importante sur l'application *ObjectsLogger*, les deux personnes seront tout de même amenées à développer par la suite sur l'application *gestOrg*.

Une fois tous ces développements réalisés, le développement du *portail App* et de l'application pilote pourra réellement débuter, on effectuera tout d'abord le développement du socle technique avec comme priorité les composants liés à la couche de présentation (interface graphique responsive, charte graphique, gestion des formulaires, etc.) et les composants liés à la couche de logique (authentification, gestion des droits, validation des données, etc.). Une fois les composants de base réalisés et avec une première version du socle technique, le développement de l'application pilote pourra commencer et évoluer conjointement avec le socle technique du *portail App*.

Concernant la couche de gestion des données du socle technique du *portail App*, comme celui-ci est déjà très bien géré par l'ORM Doctrine 2, la seule valeur ajoutée du socle technique sera de mettre en place la couche de traçabilité en interfaçant *ObjectsLogger* avec le *portail App*. De plus comme cet aspect n'est pas bloquant pour la réalisation du projet, on pourra utiliser ce temps de développement comme levier en cas de retard.

4.5 Développement du projet

4.5.1 Composants PHP distribuables

Nous avons développé plusieurs composants PHP distribuables qui sont maintenant réutilisables par une application tierce quelconque. L'intégralité de ces composants sont utilisés par le *portail App*. Chaque composant dispose également d'une classe dédiée aux tests unitaires du composant. La figure 4.20 montre l'exemple de la structure d'un composant PHP.

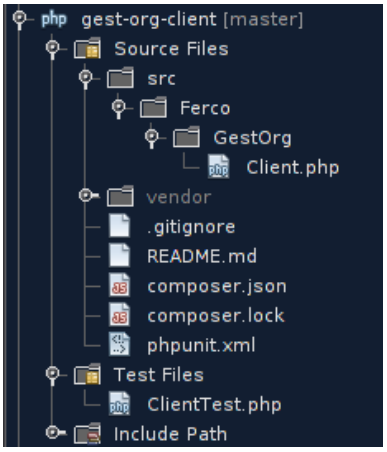


FIGURE 4.20 – Exemple d'un composant

Cette structure respecte un standard que nous avons défini en suivant les bonnes pratiques :

— Un dossier src : ce dernier contient dans cet exemple qu'une seule classe "Client.php".
— Le dossier "vendor" contient l'ensemble des dépendances de ce composant (ces dépendances seront d'ailleurs répercutées sur le projet qui va utiliser ce composant). Ce dossier est généré automatiquement par le logiciel Composer et le développeur ne doit pas intervenir dans son contenu.
— Un dossier tests : ce dossier contient les classes qui vont permettre de tester notre composant, les tests unitaires sont réalisés à l'aide de l'outil PHPUnit, dans cet exemple on retrouve le fichier ClientTest.php.
— Un fichier README.md qui permet de documenter le composant, cette manière de documenter présente l'avantage de pouvoir être consultable en ligne depuis le dépôt GIT, ainsi que d'avoir une documentation intégrée dans les fichiers du composant PHP.
— Un fichier composer.json qui définit les dépendances du composant.
— Un fichier *phpunit.xml* afin de définir les règles des tests unitaires.

Nous allons maintenant présenté rapidement les composants PHP qui ont été développés dans le cadre de notre projet.

Le composant de la gestion des dates (awesome-date)

Il s'agit d'une librairie en PHP qui permet de travailler facilement avec les dates. Elle étend la librairie standard DateTime de PHP, en apportant des méthodes supplémentaires permettant de simplifier l'utilisation des dates. Elle apporte également la prise en charge des jours ouvrés, en prenant en compte les weekends et les jours fériés. (Par exemple : Ajouter 2 jours ouvrés à la date voulue).

```
1  // Exemple d'utilisation du composant de gestion des dates
   <?php
3  // Création de dates
   $date = new Date('now');
5  $date = new Date('next monday');

7  // Manipulation de dates
   $tomorrow  = Date::now()->add('1 day');
9  $date = Date::now()->add('P2Y4DT6H8M');

11 // Ajout de 2 jours ouvrés
```

```
$date = Date::now()->addWorkingDay('2');
?>
```

Le composant de la gestion des utilisateurs (user-bundle)

Ce composant est un bundle distribuable qui vise à apporter la prise en charge de la gestion des utilisateurs qui provient de l'application *gestOrg* à une application Symfony 2. Il est donc possible simplement d'utiliser les données de la *gestOrg* pour authentifier les utilisateurs de notre application.

Le composant de communication avec la *gestOrg* (gest-org-client)

Ce composant est une librairie qui va fournir un moyen simple de dialoguer avec la *gestOrg* en utilisant ses API. Ce composant sera défini en tant que service dans notre *portail App*, de cette manière il sera très facile d'utiliser les API que propose l'application *gestOrg*.

```
<?php
// Exemple d'utilisation d'une API de la gestion de l'organisation
$client = new Client("CHEMIN_VERS_BASEURL_API", "password");
$client->call(Client::PERSONNE_FIND_API, array("id" => 2510));
?>
```

Le composant de communication avec SAP (php-saprfc)

Ce composant va proposer des méthodes pour appeler les fonctions du système SAP. Il sera utilisé par nos trois applications dans le but de réaliser la communication avec le système SAP. Voici ci-dessous un exemple simple de l'utilisation de ce composant.

```
// Exemple d'appel d'une fonction SAP avec l'aide de notre composant
<?php
  $saprfc = new PhpSaprfc($tabParams); //$tabParams contient la configuration vers le
    serveur SAP
  // $tabParamsFonction est un tableau avec les paramètres de la fonction
  $resultat = $saprfc->callFunction("NOM_DE_LA_FONCTION", $tabParamsFonction);
?>

```

Le composant de l'authentification automatique (php-ntlm)

Php-ntlm va avoir pour but de délivrer un service qui va gérer des requêtes NTLM avec le client pour retrouver l'identifiant utilisé lors de la connexion sur le domaine Windows. Ce composant est utilisé afin de réaliser une connexion SSO (single sign on) ce qui évite à l'utilisateur de devoir se connecter une nouvelle fois sur une application. Ci-dessous, un exemple d'utilisation du composant :

```
// Exemple pour récupérer le login de la personne
<?php
$login = PhpNtlm:getUsername();
// $login contient par exemple "dillenschneider_v", avec ce login on peut ensuite identifier
// automatiquement la personne et retrouver ses informations depuis l'application.
?>
```

Le composant de la génération des fichiers PDF (pdf-builder)

Pdf-builder va avoir pour but de générer des fichiers PDF en respectant un format de sortie pour la société FERCO. Le fichier PDF sera généré à partir de contenu HTML. Il est à noter que ce paquet va dépendre de la librairie PHP TCPDF. Ci-dessous, un exemple d'utilisation du composant :

```
    // Exemple pour réaliser l'export d'un fichier PDF
2   <?php
    use Ferco\PdfBuilder\PdfBuilder;
4
    $pdf = new PdfBuilder("Export de demande d'investissement","Vincent");
6   $html = "<html><body><p>Hello World!!</p></body></html>";

8   $pdf->writeHTMLPage($html);
    $pdf->Output('test.pdf', 'I');
10
```

4.5.2 L'application *gestOrg*

Le développement de l'application gestion de l'organisation est séparée en deux parties : d'un côté le backend dont le but est de s'occuper de la gestion des données et de fournir des API aux autres systèmes, et de l'autre côté on retrouve le frontend qui va gérer toute la partie présentation (client riche), afin que le gestionnaire puisse effectuer ses actions en toute ergonomie.

4.5.2.1 Le backend

Dans l'approche de l'architecture MVC , le backend va gérer les couches Modèle et Contrôleur de l'application. Il est développé avec le framework Symfony 2 et ce dernier respecte les conventions du framework en appliquant la structure des fichiers et des répertoires préconisée. L'ensemble du code source sera crée dans le bundle "GestionOrganisationBundle" comme le montre la figure 4.21.

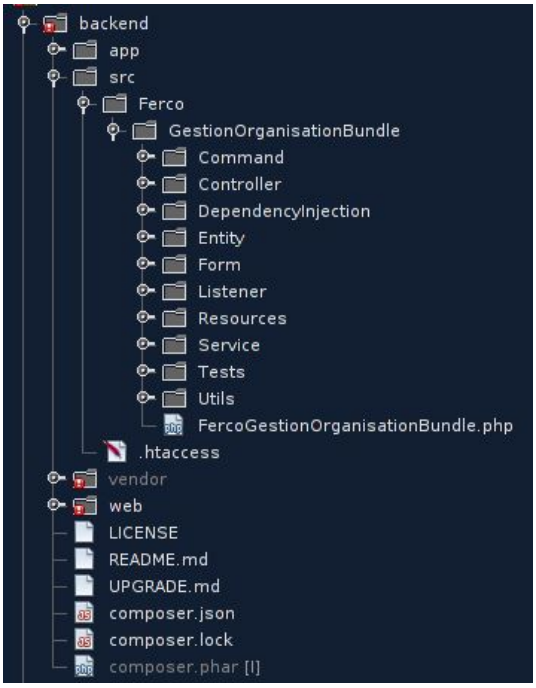


FIGURE 4.21 – Structure de l'application

Les entités

Les entités sont placées dans le répertoire "Entity". Chaque entité est représentée par une classe PHP qui va définir les propriétés de l'objet métier, ainsi que ses méthodes d'accès. On retrouve ainsi dans ce dossier les fichiers suivants :

— AutorisationApplicatif.php
— AutorisationGestion.php
— CompteUtilisateur.php
— Groupe.php
— Personne.php
— RoleApplicatif.php
— RoleGestion.php
— Type.php

La classe PHP est analysée par l'ORM Doctrine 2 via un système d'annotation, il est alors possible de préciser comment chaque attribut sera géré par l'ORM. Une commande de mise à jour de la base de données est fourni par l'ORM afin de permettre la génération et la synchronisation d'une base de données depuis nos classes PHP.

Les contrôleurs

Un contrôleur est disponible pour chaque objet métier dans le répertoire "Controller". Le contrôleur va fournir les API pour interagir avec les entités citées précédemment.

La commande de synchronisation

La commande de synchronisation qui va pouvoir effectuer la mise à jour de la base de données local se trouve dans le répertoire "Command" de l'application. Cette dernière est exécutable une fois

positionnée dans le dossier backend via la commande suivante "php app/console ferco :organisation :update". Une fois exécutée, la synchronisation des données de l'application sera effectuée avec le système SAP-HR (données salariés) et l'Active directory (données utilisateurs).

4.5.2.2 Le frontend

Le frontend de l'application est réalisé à l'aide du framework ExtJs. La structure de l'application respecte également les bonnes pratiques du framework. La figure 4.22 présente la structure du frontend.

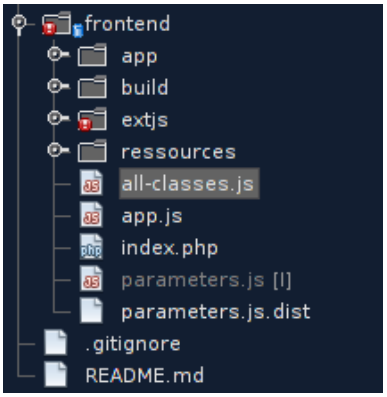


FIGURE 4.22 – Structure du frontend

Le dossier app

Le dossier app contient l'ensemble du code source de l'application avec les éléments suivants :

— Les contrôleurs : ils vont permettre de définir les actions associées aux événements de l'application, par exemple le clic sur un bouton.
— Les modèles : ils vont définir les entités de l'application, toutes les entités du backend seront présentes ici.
— Les store : ils vont servir à fournir les données à l'application, par exemple le store "personne" va avoir pour responsabilité d'utiliser un proxy AJAX pour récupérer les données depuis l'API du backend afin de les servir dans l'application frontend à travers le store. De la même manière il sera chargé d'effectuer la synchronisation des données : si une personne est ajoutée dans le store, une requête AJAX est envoyée à une API du backend afin que les modifications se répercutent dans la base de données.
— les vues : dans cette partie on retrouvera la partie présentation de l'application (tableaux, formulaires, boutons, etc.).

Le fichier app.js

Le fichier app.js a pour rôle d'amorcer l'application, il va enregistrer les contrôleurs de l'application, définir les variables globales et lancer le conteneur qui va afficher la première page de présentation.

Le fichier parameters.js

Dans ce fichier on retrouve les paramètres qui vont être utilisés par le fichier d'amorçage app.js.

Le fichier index.php

C'est le fichier qui va être délivré par le serveur Web au client et qui va inclure le fichier d'amorçage app.js.

L'interface graphique

Comme annoncé dans nos besoins, l'interface de gestion doit être simple et intuitive. La figure 4.23 montre un aperçu de l'interface graphique.

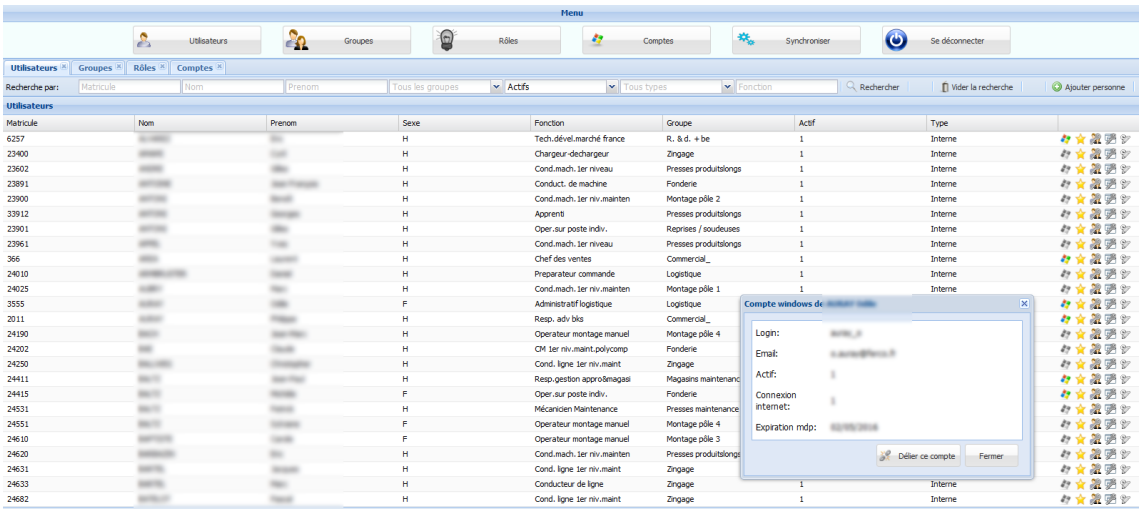


FIGURE 4.23 – Interface graphique de l'application

Le haut de la page regroupe les boutons qui ouvrent des onglets où il sera possible de gérer les données de la section choisie. Les tableaux sont présentés sous la forme de grille avec une interface de recherche au-dessus, il est également possible de trier les colonnes en cliquant sur l'intitulé voulu. Chaque lien au niveau de l'interface graphique va ouvre une popup avec la possibilité de visualiser ou éditer les données sélectionnées.

4.5.2.3 La sécurité des services Web

Certaines API seront publiques et ne nécessiteront pas de sécurité particulière. Les API privées seront elles sécurisées via une authentification WSSE (voir partie 3.3.2.9, paragraphe WS-Security), cette authentification sera implémentée par un RequestListener au niveau du backend de l'application. Ce composant va intercepter toutes les requêtes HTTP entrantes et vérifier si une signature valide est présente au niveau de leurs entêtes.

4.5.3 L'application *ObjectsLogger*

L'application *ObjectsLogger* est entièrement développée avec le framework Symfony 2 et respecte ses conventions. L'ensemble du code source de l'application est développé dans le bundle "ObjectsLoggerBundle". La figure 4.24 illustre la structure de l'application *ObjectsLogger*.

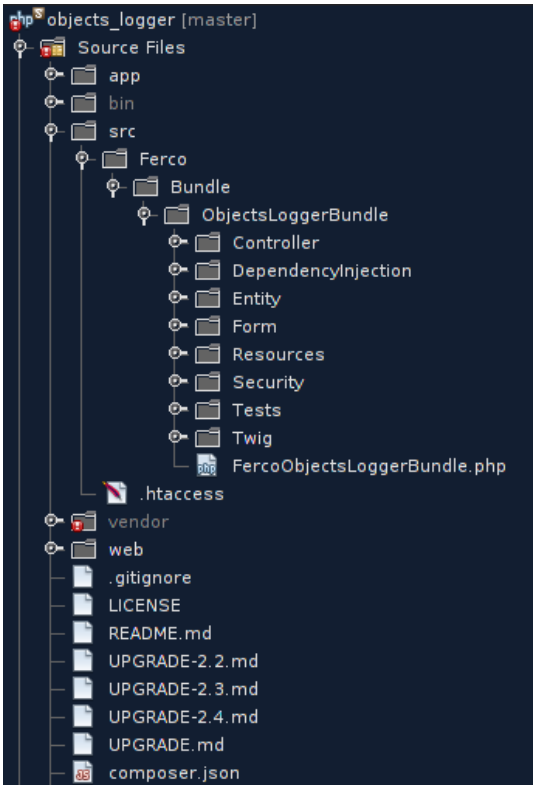


FIGURE 4.24 – Structure de l'application *ObjectsLogger*

Les entités

Toujours afin de respecter le standard préconisé par le framework Symfony 2, les entités de l'application sont placées dans le répertoire "Entity". On retrouve dans ce dossier les fichiers suivants (où chaque fichier PHP contient une entité) :

— Application.php
— Classe.php
— Log.php

Les contrôleurs

Les contrôleurs de l'application sont disponibles dans le répertoire "Controller". On retrouve dans ce répertoire, un contrôleur pour chaque entité où nous allons retrouver les services Web correspondants :

— ApplicationController.php
— ClasseController.php
— LogController.php

Le formulaire

L'alimentation des données se fera à l'aide des services Web qu'utiliseront les applications tierces, il n'y aura donc aucun formulaire de saisie pour insérer des informations dans la base de données. Cependant un formulaire de recherche sera disponible pour que le développeur puisse rechercher

facilement les *messages de logs* en fonction de certains critères de recherche comme l'identifiant de l'objet et l'espace de nom de la classe.

Les vues

Les vues de l'application sont disponibles dans le répertoire "Resources/views". Pour cette application nous avons développé trois vues :

— La page d'accueil de l'application
— La page principale de visualisation d'une log, cette dernière contient la vue de l'objet au format JSON, les données administratives et l'historique des *messages de logs* sur l'objet.
— La page de comparaison de deux *messages de logs* d'un même objet, ce qui permet de repérer facilement les éléments de l'objet qui ont été modifiés.

Les services Web

L'ensemble des API sont soumises aux mêmes principes que les services Web de l'application *gestOrg*, c'est à dire qu'elles sont également sécurisées via une authentification WSSE.

4.5.4 Le *portail App*

Le *portail App* est une application Web complexe développée sous le framework Symfony 2.

4.5.4.1 Le socle technique

Le socle technique du *portail App* est représenté par un bundle technique indépendant nommé "Main/LayoutBundle", ce nom a été choisi car la majorité des éléments techniques sont liés à la couche de présentation. Nous allons présentés rapidement ci-dessous les éléments qui composent le socle technique du *portail App*.

La gestion de la connexion

L'ensemble du *portail App* délivre des services à un utilisateur précis, son utilisation est donc exclusivement "privée", il est alors nécessaire de procéder à une authentification de l'utilisateur quelque soit la page Web demandée.

Cette gestion de la connexion est le premier point qui a été implémenté dans le *portail App* et il s'agit également d'un point complexe. En effet dans le cahier des charges, il a été stipulé que les utilisateurs peuvent se connecter à l'aide de trois moyens de connexion : une connexion automatique, une connexion avec un couple identifiant/mot de passe et enfin le dernier moyen de connexion est le badge du salarié.

Afin d'implémenter ces méthodes de connexion nous allons utiliser la configuration souple de Symfony 2 pour définir la partie protégée de notre application et rediriger les requêtes vers un contrôleur spécifique qui aura comme unique tâche d'orchestrer la connexion de l'utilisateur si celui-ci n'est pas déjà authentifié. Ce contrôleur utilisera le composant PHP "user-bundle" développé précédemment qui va nous fournir les données de la gestion de l'organisation pour valider les informations de l'authentification et récupérer les autorisations liées de la personne connectée (dans le but de d'obtenir les services et les ressources que cette personne peut utiliser). Une particularité qui a complexifiée

ce développement est la méthode de connexion automatique, car contrairement aux deux autres méthodes, cette dernière implique une authentification NTLM avec l'ActiveDirectory.

La figure 4.25 montre l'exemple du processus d'authentification général. La connexion automatique est d'abord initiée par le système, si ce dernier ne retrouve pas de correspondance entre le compte utilisateur Windows fournit par le navigateur (qui doit être validé par le service d'authentification du contrôleur de domaine Windows) et la base de données de la *gestOrg*, alors il est redirigé vers une page Web où il peut choisir son moyen de connexion (par identifiant ou par le badge) afin de relancer son authentification. Ce cas précis où l'utilisateur ne peut pas être authentifié de manière automatique arrive lorsque ce dernier utilise un compte générique qui n'est pas personnel et ne peut donc pas être associé automatiquement avec les informations d'un salarié.

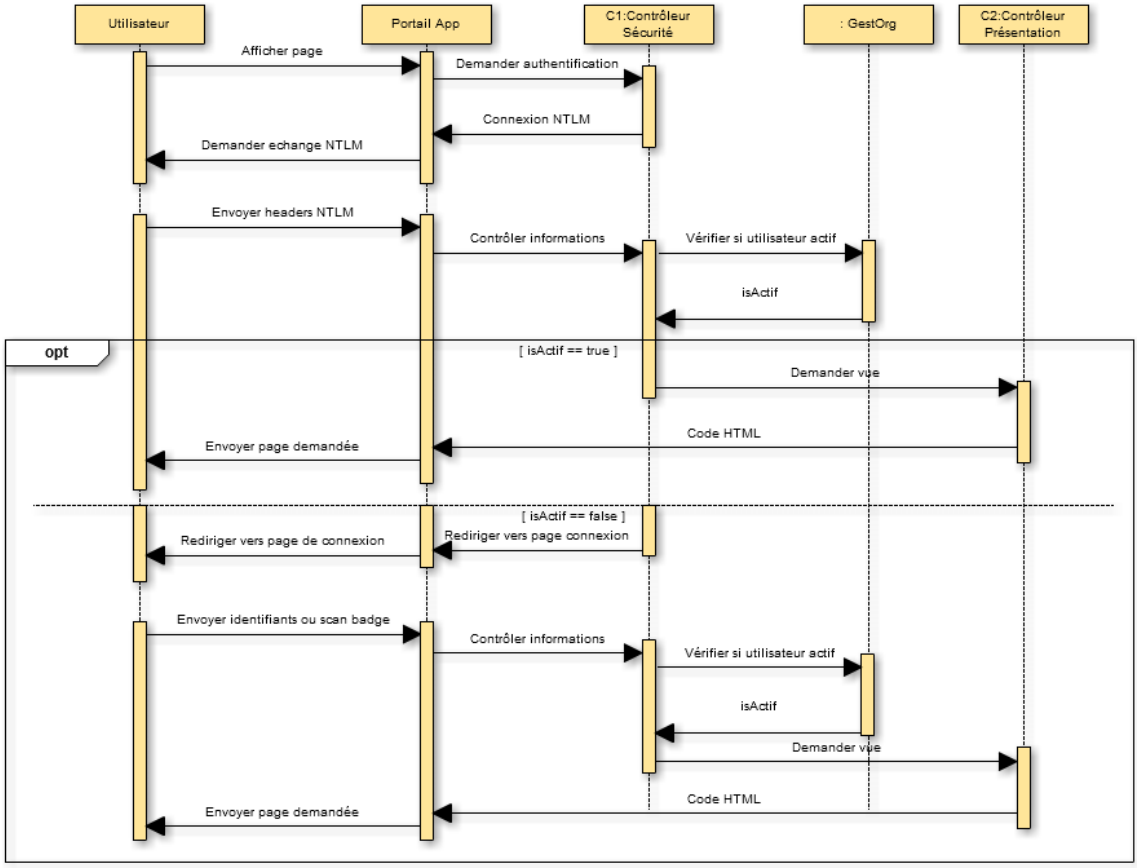


FIGURE 4.25 – Connexion de l'utilisateur

Le layout

Nous avons utilisé la technologie Twig qui est parfaitement couplée avec le framework Symfony 2 pour définir une structure basique.

Charte graphique et gestion de la mobilité

Nous n'avions pas de charte graphique imposé et après notre étude sur la question de la mobilité, nous avions choisi de partir sur l'excellente librairie Bootstrap 3. Cette librairie offre des composants de base pour traiter la question de la mobilité, par contre elle n'offre pas une solution clé en main

pour définir une charte graphique. Pour définir notre propre charte graphique, nous sommes parti d'un thème opensource basé sur Bootstrap 3 que nous avons modifié pour les besoins de l'entreprise.

Les aides de vues

Twig est une librairie qui permet non seulement de réaliser des templates (le layout) de notre application, mais elle permet aussi de faciliter le développement des vues en proposant des fonctions dédiées à l'affichage des variables. Afin d'optimiser encore plus notre développement, nous avons réalisé des extensions au sein même de notre socle technique pour fournir aux développeurs des fonctions dédiées à notre entreprise.

La figure 4.26 montre les extensions qui ont été développées.

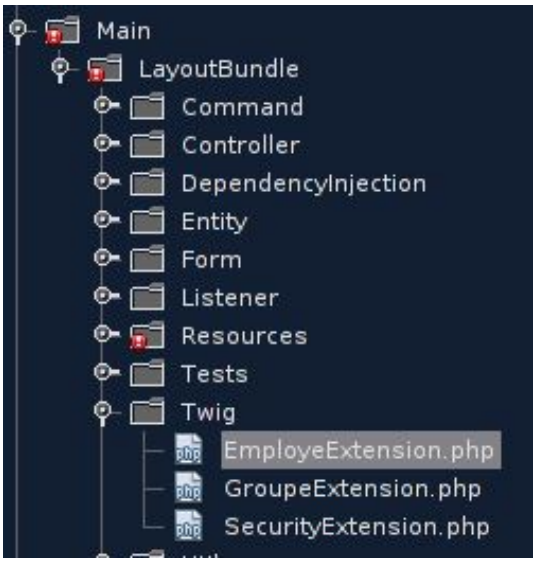


FIGURE 4.26 – Extensions Twig

Par exemple, au sein de l'extension qui concernant les employés "EmployeExtension.php", on retrouve une classe qui contient plusieurs méthodes qui permettent de définir des fonctions Twig. Par exemple, la fonction "fiche_salarie" qui permet de générer un lien qui va afficher les informations du salarié via une popup est utilisable dans un template twig de la manière suivante :

```
1       // Ce code twig va afficher un lien où un clic va
        // afficher la fiche de salarié sous forme de popup
3       <p>Salarié : {{fiche_salarie(matricule)}}</p>
```

Le résultat du clic sur le lien généré est illustré par la figure 4.27.

FIGURE 4.27 – Exemple de la fiche d'un salarié

Les formulaires Web

Les formulaires Web sont un aspect incontournable pour la gestion des données dans une application. Les formulaires sont très bien intégrés dans Symfony 2 avec des éléments qui permettent de valider le format des données facilement. Par exemple, un champ de type numérique va être automatiquement contrôlé et validé par le framework. Pour aller encore plus loin et gagner encore en efficacité, nous avons développé des types de champs personnalisés qui permettent de produire des champs spécifiques à des besoins précis. Nous pouvons citer notamment le champ personnalisé "employe_autocomplete" qui permet d'afficher à l'écran un champ de saisie avec une autocomplétion avec tous les salariés actifs de l'entreprise. Lors de la saisie d'une donnée dans ce champ (de minimum trois caractères), le système récupère la liste des salariés actifs dont le nom débute avec les caractères saisies et affiche cette liste à l'utilisateur qui n'a plus qu'à choisir la personne voulue. Une fois que ce champ est soumis avec le formulaire, le backend du système va vérifier automatiquement si la valeur contenue dans le champ est en accord avec la liste des salariés actifs de l'entreprise et affiche un message d'erreur en cas de problèmes. La figure 4.28 montre les résultats après avoir saisi la chaîne de caractères "dill" dans le champ personnalisé.

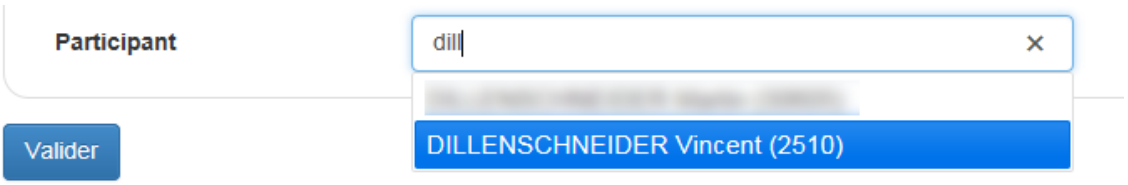


FIGURE 4.28 – Champ d'autocomplétion sur le nom d'un salarié

La normalisation des données

Afin d'afficher correctement les données à l'écran de l'utilisateur ou de les intégrer proprement dans le système à partir des formulaires Web, nous avons recours à la normalisation des données. Par exemple dans le cas du champ "employe_autocomplete", la donnée système est un entier (le matricule) et la donnée affichée dans le champ est une chaîne de caractères (nom, prénom et matricule). Pour réaliser automatiquement cette transformation nous avons mis en place un "data transformer" qui est associé au type de champ, ce dernier va alors automatiquement convertir la chaîne de caractères saisie par l'utilisateur en un entier et inversement pour l'affichage.

Gestion des *messages de logs*

Au niveau de la gestion des *messages de logs*, nous pouvons distingué les messages du système et les messages liés aux objets métiers. Les messages du système sont automatiquement gérés par le framework Symfony 2, ce dernier enregistre les événements survenus dans un fichier dev.log ou prod.log (en fonction de l'environnement d'exécution). Les messages métiers sont eux automatiquement gérés par une classe que nous avons développé et qui enregistre un écouteur d'événement du framework Doctrine 2. Cet écouteur permet que notre classe soit systématiquement instanciée lors de la création ou modification d'entités par le framework Doctrine 2. Cette classe va ensuite dialoguer avec notre application tierce *ObjectsLogger* qui permet de sauvergarder l'état de nos objets métiers. Concrètement, à chaque modification des données d'une entité gérée par Doctrine 2, l'objet sérialisé au format JSON est envoyé à l'API de *ObjectsLogger*, ce dernier n'a plus qu'à enregistrer ce nouvel état dans la base de données.

L'export des données

Nous avons principalement deux exports au niveau du PAE, les exports PDF pour les impressions et les exports Excel pour tout ce qui va être extraction de la base de données.

Concernant les extractions PDF, nous avons réalisé un composant pdf-builder qui utilise la librairie TCPDF (librairie qui permet de gérer les PDF avec le langage PHP). Ce composant définit le layout général du PDF (header et footer), l'ensemble du PAE dispose alors d'une gestion simple pour produire un fichier PDF. En effet, il suffit simplement dans le cadre d'un export PDF de renseigner uniquement le contenu du rapport, ce qui peut être effectué à l'aide d'une vue twig avec comme résultat un rendu HTML qui sera interprété le composant puis transmis au navigateur sous la forme d'un fichier PDF. Cette solution pour produire des fichiers PDF est très efficace par rapport au portail Intranet, ce dernier devait utiliser des API pour écrire du contenu PDF, le fait de produire un fichier PDF depuis du code HTML permet de garder une homogénéité dans l'application (la très grande majorité des vues Twig utilisent du code HTML) et de réduire la complexité.

Pour les extractions Excel, le principe est différent, le PAE possède une dépendance avec un composant fournit par la communauté "mewesk/twig-excel-bundle", ce dernier offre la possibilité de personnaliser le système de routing pour indiquer quel format de retour est attendu (csv, xls, xlsx) ainsi que rajouter des directives à Twig permettant de générer facilement un fichier Excel. L'exemple ci-dessous illustre une vue Twig qui génère un fichier Excel.

```
1 {% xlsdocument %}
    {% xlssheet 'Worksheet' %}
3     {% xlsrow %}
        {% xlscell { style: { font: { size: '18' } } } %}Values{% endxlscell %}
5     {% endxlsrow %}
      {% for value in data %}
```

```
7         {% xlsrow %}
            {% xlscell %}{{ value }}{% endxlscell %}
9         {% endxlsrow %}
      {% endfor %}
11  {% endxlssheet %}
{% endxlsdocument %}
13
```

4.5.5 *portail App* - développement de l'application pilote

L'application pilote de la gestion des déplacements prend la forme d'un bundle Symfony 2 qui est intégré au *portail App*. L'ensemble du code source qui compose cette application est situé dans le bundle "GestDepBundle". La figure 4.29 montre la structure de ce bundle.

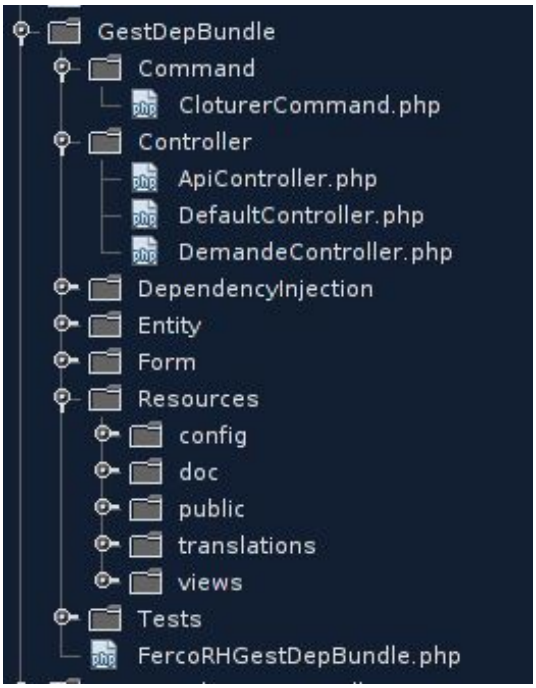


FIGURE 4.29 – Structure du GestDepBundle

Les entités

Les entités sont situées dans le dossier "Entity". On retrouve dans ce dossier les classes qui reflètent nos objets métiers exprimés dans le cahier des charges :

— Demande.php
— Hotel.php
— Participant.php
— Status.php
— Transport.php
— TypeDemande.php
— TypeTransport.php
— VoitureLoc.php

Les contrôleurs

Nous disposons pour cette application de trois contrôleurs :
— ApiController.php : Contient une API qui renvoie les déplacements des salariés en fonction d'une sélection de dates. Aucune couche de sécurité n'est à prévoir car il s'agit d'une API publique.
— DefaultController.php : Contient les rapports et les extractions de l'application.
— DemandeController.php : Contient tout ce qui est en lien avec la gestion de la demande : consultation, création, modification, etc.

Les vues

Les vues Twig sont stockées dans le répertoire "views". On retrouve dans ce dossier des vues pour générer des contenus de mails, des pages Web, des reportings, etc.

La commande de clôture

Située dans le dossier "Command", la commande de clôture permet au système de transmettre les informations des déplacements au système SAP-HR. Cette commande sera lancée la veille de chaque déplacement à 20h15 pour que les données soient dans le système SAP-HR le lendemain afin que les gestionnaires RH puissent traiter leurs anomalies.

4.6 Phase de test

4.6.1 Les tests unitaires des composants

Afin de garantir la fiabilité des composants PHP développés, nous avons profité de ce projet afin de mettre en place l'outil *PHPUnit* qui permet de réaliser des tests unitaires en PHP.

Comme nous l'avons mentionné dans la partie 4.5.1, un composant va respecter une structure. Au sein de cette structure, on retrouve un dossier tests qui va contenir les fichiers de tests unitaires, ainsi qu'un fichier *phpunit.xml* qui va inclure des directives pour paramétrer l'outil *PHPUnit*.

Afin d'illustrer comment s'articule un test sous *PHPUnit*, nous allons prendre l'exemple d'un composant et voir les actions qu'il est nécessaire d'effectuer pour réaliser nos tests unitaires.

4.6.1.1 Test du composant *gest-org-client*

Un pré-requis pour que les tests unitaires fonctionnent est d'avoir un environnement avec l'outil *PHPUnit* correctement installé. Pour savoir si c'est le cas, il suffit de taper la commande *phpunit –version* dans un terminal et de constater si la commande affiche bien la version du logiciel.

Nous allons maintenant présenter les trois étapes nécessaires pour réaliser nos tests unitaires.

Etape 1 : La classe de test

La classe de test *ClientTest* se trouve dans *tests/ClientTest.php*. Cette dernière définit plusieurs éléments pour le bon déroulement des tests :
— l'initialisation du cadre de test via la méthode *setUp*,
— un lancement de test pour chaque méthode qui commence par le mot *test*, par exemple la méthode *testFindPersonne*,

— Le nettoyage de l'environnement une fois que l'ensemble des tests ont été déroulés via la méthode *tearDown*.

La figure 4.30 montre un aperçu du contenu de la classe *ClientTest*. On remarquera que les tests sont en réalité des opérations d'assertions qui sont délivrées par la classe *PHPUnit_Framework_TestCase*. Dans notre exemple, on remarque que la méthode *testFindPersonne* contient trois assertions.

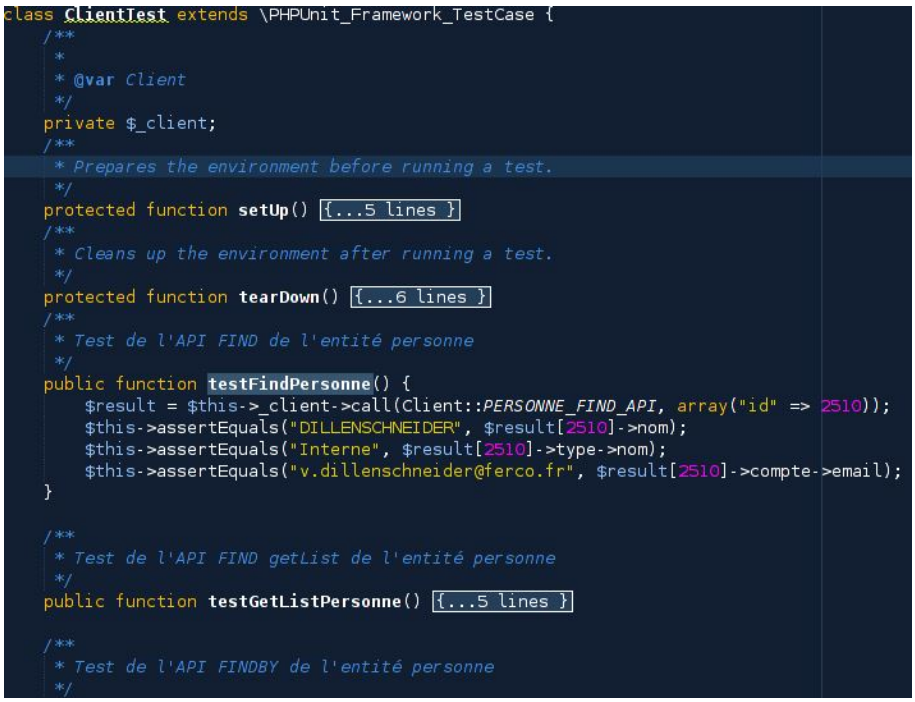


FIGURE 4.30 – Aperçu de la classe *ClientTest*

Etape 2 : Paramétrer *PHPUnit* pour le composant

Il est possible de lancer les tests de plusieurs manières avec l'outil *PHPUnit*, la plus pratique reste de définir un fichier la configuration du logiciel et de lancer simplement la ligne de commande *phpunit* à la racine du projet (ou d'utiliser une fonctionnalité de l'IDE).

La figure 4.31 présente le contenu du fichier *phpunit.xml* qui est utilisé par *PHPUnit*.

FIGURE 4.31 – Aperçu du fichier *phpunit.xml*

Etape 3 : Lancer les tests

Afin de lancer les tests, nous pouvons utiliser une fonctionnalité intégrée à Netbeans qui permet d'obtenir une synthèse des tests unitaires.

La figure 4.32 illustre le lancement des tests unitaires via l'IDE Netbeans.

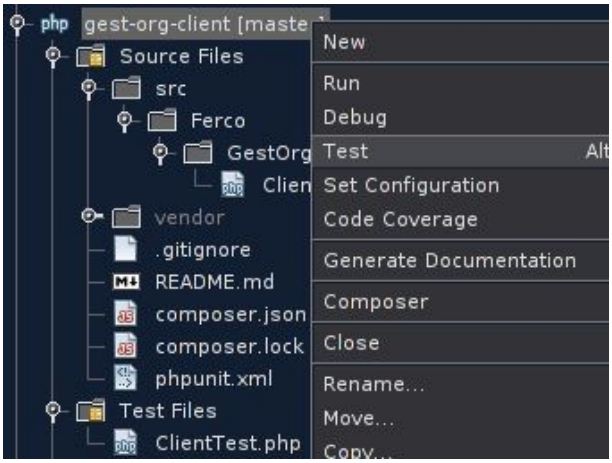


FIGURE 4.32 – Lancement des tests unitaires via Netbeans

La figure 4.33 présente le résultat des tests unitaires. On peut remarquer que l'on retrouve à l'écran les différentes erreurs, ainsi que plusieurs informations utiles comme le temps d'exécution du programme, le nombre de tests qui ont échoués, etc. L'IDE est pratique pour le lancement des tests, car hormis un visuel plus conviviale que la ligne de commande, il est également possible de relancer uniquement les tests en échec en appuyant simplement sur le bouton adéquat.

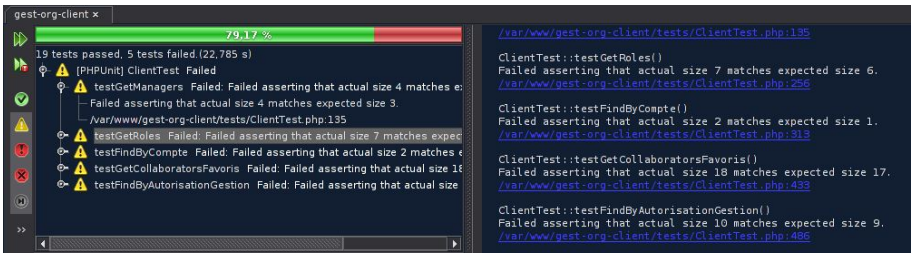


FIGURE 4.33 – Résultats des tests unitaires

4.6.2 Les scénarios de tests

En dehors des tests unitaires, nous avons dans le cadre du projet réalisé des scénarios de tests. Contrairement aux tests unitaires où nous avons mis en place un nouvel outil, pour effectuer les scénarios de tests, nous avons respecté la procédure de gestion de projets de la société FERCO et par conséquent nous avons utilisé les standards définis par l'entreprise. Les scénarios de tests sont réalisés à l'aide d'un fichier Excel où l'on retrouve les différents cas de tests afin de valider les fonctionnalités de l'application. Pour chaque cas, on précise le résultat attendu et on indique la date de validation du test lors de son déroulement. La figure 4.34 présente un aperçu du scénario de tests de l'application de la gestion des déplacements.

PT_GESTION_DEPLACEMENT

OBJECTIFS					GLPI n° :	11260	
					Testeur :	Vincent DILLENSCHNEIDER / Nicolas ALVAREZ	
					Correcteur :	Vincent DILLENSCHNEIDER / Nicolas ALVAREZ	
CORRECTION			Résultat attendu	Risque	Validé le	Description des anomalies	Corrigé le
	Ajouter Demande						
		Un participant n'existe pas ou est inactif	Le système ne sauvegarde pas la demande et affiche un message d'erreur		13/03/2014	Le système filtre et ne propose plus les salariés externes et inactifs	
		Erreur format de données	Le système ne sauvegarde pas la demande et affiche un message d'erreur		13/03/2014	Le système refuse la saisie et indique les endroits où les erreurs de format sont survenues	
		Erreur incohérence de données	Le système ne sauvegarde pas la demande et affiche un message d'erreur				
		Le responsable de la personne n'est pas trouvé	Le système ne sauvegarde pas la demande et affiche un message d'erreur				
		Détermination de la validation par défaut					
		Le demandeur est le gestionnaire de déplacement ou le président	Le système sauvegarde la demande dans tous les cas et **ne lance aucune validation.** L'état de la demande est "validée".		14/03/2014	Details dans onglet gestionnaire / president	
		Le salarié effectue une demande de déplacement pour lui-même	Le système sauvegarde la demande et effectue une demande de validation. L'état de la demande est "en attente de validation"		17/03/2014	CAS VD --> validation L IMM	

FIGURE 4.34 – Aperçu du scénario de tests de l'application gestion des déplacements

4.6.3 Validation des utilisateurs

Après la validation de ces scénarios de tests, des réunions avec différents panels d'utilisateurs (cadres, employés et cadres dirigeants) ont été organisées dans le but de valider l'ergonomie générale du *portail App* et de réaliser un test d'utilisabilité comme préconisé dans [NOGIER, 2008]. Suite à ces réunions, nous avons déterminé une période de quinze jours où ces utilisateurs pouvaient tester et valider le fonctionnement de l'application pilote.

A la fin de cette période de tests, le *portail App* a été mis à jour avec les modifications pertinentes qui ont été demandées par les utilisateurs et validées par le groupe de travail du projet.

4.7 Modification des processus des développements informatiques

L'amélioration de l'efficacité des développements informatiques était l'un des objectifs de ce projet. La veille technologique qui a été réalisée dans le cadre du projet a permis de remettre en cause certains processus dans le cadre du développement Web à FERCO.

4.7.1 Gestion des tests

Avant ce projet, la gestion des tests se basait uniquement sur des tests de scénarios directement sur le navigateur Web, aucun tests de composants indépendant n'avait eu lieu. Ce projet a permis de mettre en place l'excellent outil *PHPUnit* couplé avec l'IDE Netbeans, cette association a permis de réaliser facilement des tests unitaires sur les composants PHP distribués qui ont été développés. La gestion des tests unitaires apporte plus de sécurité lors de la maintenance et de l'évolution des applications, cette nouvelle gestion des tests est maintenant devenue une bonne pratique des développements à FERCO.

4.7.2 Gestion des codes sources

La phase d'étude du projet nous a donné la possibilité de réfléchir sérieusement sur la gestion des codes sources de nos applications, c'est une partie souvent négligé dans le développement mais une bonne gestion des versions facilite beaucoup le développement et le suivi des modifications. Notre nouvelle méthodologie avec la mise en place de GIT et des branches est vraiment intéressante et a

permis un développement plus souple et plus serein. Après avoir expérimenté un certain nombre d'avantages avec ce système (passage d'une branche à l'autre, gestion des conflits simple et fusion de branches), il est impensable de revenir en arrière sans perdre de la qualité et de l'efficacité dans le travail du développeur.

4.8 Documentation

Tout au long du projet, nous avons réalisé trois types de documentation.

4.8.1 La documentation technique

La documentation technique va permettre au développeur de maintenir simplement les applications développées. Comme cette documentation vise le développeur, il paraît plus judicieux de l'attacher directement aux sources du projet, cela évitera les problèmes classiques de localisation de la documentation. La première documentation technique sera les commentaires et surtout la PhpDoc (un système de commentaires structuré, inspiré de la JavaDoc) qui va permettre d'exporter une documentation complète. La seconde documentation technique était l'écriture systématique d'un fichier Markdown (extension .md) qui permet d'écrire du HTML de façon raccourcie c'est à dire sans balisage, ce fichier (README.md) est placé systématiquement à la racine de chaque projet PHP, décrivant de façon générale le développement qui a été fait. Par exemple Github utilise ce système pour documenter les projets que ce dernier héberge, dans notre cas nous avons utilisé un logiciel Web nommé GitList qui permet via un navigateur Web de visualiser l'ensemble des dépôts de codes sources tout en affichant cette documentation au format HTML (voir la figure 4.35).

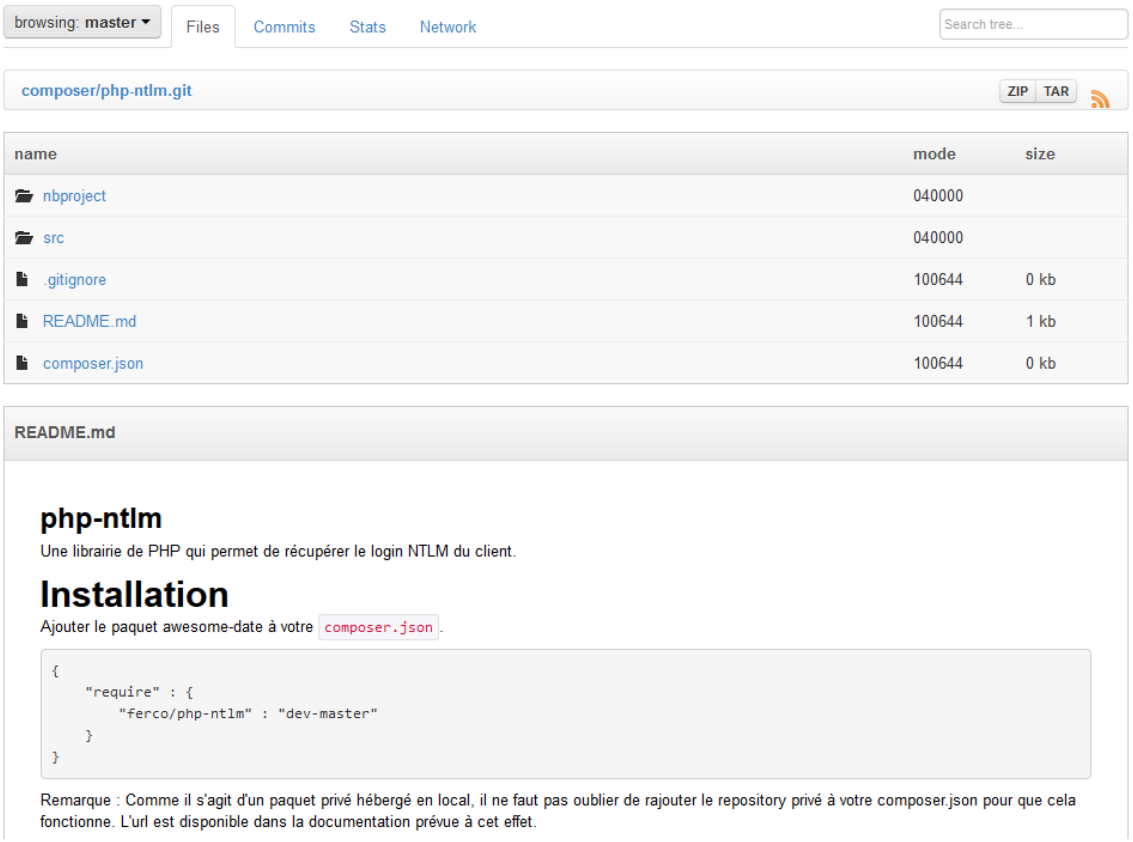


FIGURE 4.35 – Gitlist

4.8.2 La documentation des services Web

Ce projet a permis de lancer FERCO dans une démarche SOA, c'est à dire ouvrir les fonctionnalités des applications à l'aide de services structurés. Afin de respecter et de diffuser clairement l'ensemble des services Web disponibles dans le système d'informations, nous avons développé une application Web "Annuaire d'API" qui centralise la documentation des différents services Web de l'ensemble de nos applications en présentant le chemin d'accès et la description du service. La figure 4.36 montre l'exemple de la documentation d'une API au sein de l'application Web.

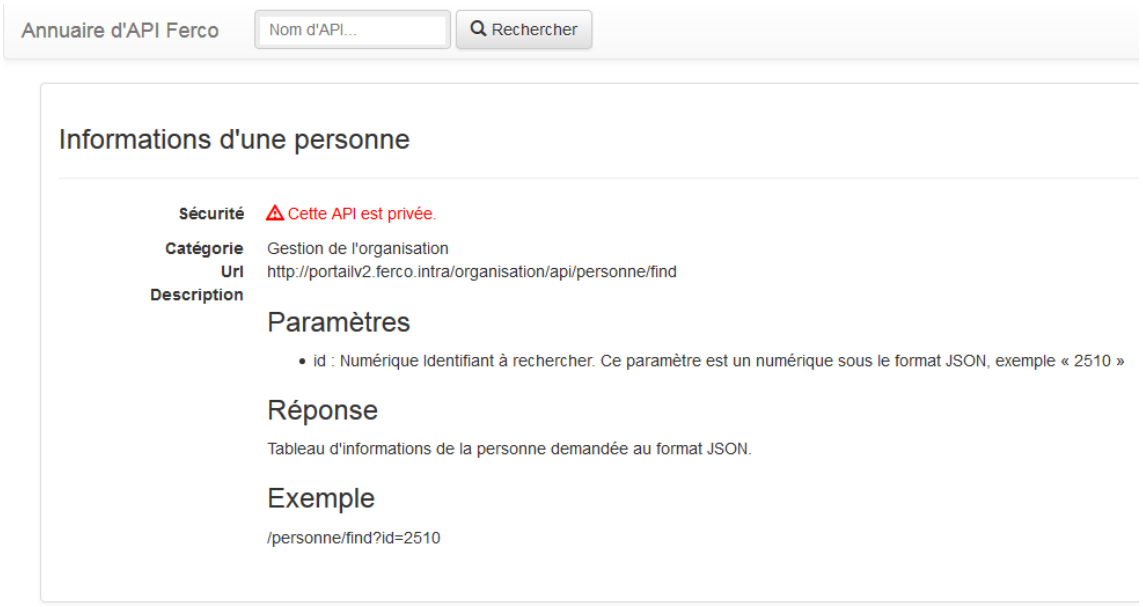


FIGURE 4.36 – Exemple de documentation d'une API

4.8.3 La documentation utilisateur

Les manuels utilisateurs des applications Web développées (comme la gestion des déplacements par exemple) sont accessibles en ligne depuis l'interface Web du *portail App*. Sur le haut de page de chaque application, une icône propose de télécharger cette documentation sous la forme d'un fichier de format PDF qui présente le fonctionnement de l'application concernée.

La figure 4.37 illustre ce principe, on retrouve l'icône d'aide en haut de l'écran à droite. Lors du clic sur cette icône, un fichier PDF est téléchargé puis ouvert dans le navigateur Web.

FIGURE 4.37 – Exemple de la documentation utilisateur

4.9 Déploiement du projet

Une fois que le projet a été validé, son déploiement a été réalisé en plusieurs étapes. Comme les composants PHP développés sont stockés via notre système de gestion de versions, il n'y a aucune action à réaliser de ce côté là, la première étape a donc été de mettre en ligne et configurer les applications tierces qui sont *ObjectsLogger* et la *gestOrg* sur notre serveur applicatif. La seconde étape a été de mettre en ligne le *portail App* et avec lui l'application pilote, cette étape a nécessité la configuration et la mise en place des interfaces entre le *portail App* et les systèmes qui vont dialoguer avec ce dernier (SAP, *gestOrg*, *ObjectsLogger*, etc.).

5 Formation et communication

5.1 Formation des utilisateurs

Avant la mise en production du *portail App* et de l'application pilote, nous avons réfléchi sur la méthode de formation des utilisateurs, en effet nous retrouvons dans l'entreprise différents profils de personnes avec des niveaux de connaissances informatiques hétérogènes, on peut par exemple citer le cadre qui va utiliser des logiciels de gestion et des outils bureautiques très régulièrement et un ouvrier sur machine qui ne va utiliser l'outil informatique que rarement. C'est dans ce contexte qu'il a été décidé d'organiser plusieurs sessions de formation afin de présenter le nouveau portail applicatif ainsi que l'application pilote. Ces formations ont principalement visées les plus grands utilisateurs de l'application pilote, les correspondants informatiques (personnes ayant des compétences informatiques plus avancées et qui font office de relais au sein de chaque service de l'entreprise) et les responsables de service.

Comme la prise en main du nouveau PAE est simple, il n'a pas été jugé utile de former l'ensemble du personnel, nous avons donc communiquer aux correspondants informatiques de l'entreprise qu'ils avaient la charge de répondre aux questions basiques concernant l'utilisation du logiciel et de transférer au service informatique toutes les questions plus avancées.

Cette décision de ne former qu'une partie des utilisateurs de l'entreprise était pertinente, cela a en effet permis d'éviter de mobiliser trop de ressources en s'appuyant sur les utilisateurs sélectionnés qui sont des personnes avec des connaissances informatiques suffisantes pour aider les autres collaborateurs sans toutefois déformer l'information qui a été transmise. Avec cette démarche, l'entreprise gagne sur plusieurs points : on responsabilise des personnes en leur donnant un rôle de référent et de formateur sur l'outil et on gagne du temps de formation en évitant des démarches d'organisation (réservation de salle et mobilisation de plusieurs personnes).

5.2 Communication de la mise en production

Au sein de l'entreprise FERCO, la communication interne globale est réalisée par un document qui se nomme "ForceInfo" et dont la diffusion est hebdomadaire. Il s'agit dans les faits d'un fichier PDF qui est diffusé via différents canaux de distribution : mail à l'ensemble de l'entreprise, un article dans l'Intranet et le document imprimé disponible en consultation dans plusieurs endroits de la société.

Ce document représente la communication officielle de l'entreprise et touche l'ensemble des salariés, nous avons donc utiliser ce support pour communiquer la mise en production de l'application de la gestion des déplacements (et par conséquent du *portail App*). Le document qui a été utilisé est disponible en annexe E du mémoire

6 Exploitation des résultats

6.1 Bilan de la réalisation

Nous avons réussi à mettre en place un PAE d'une qualité très satisfaisante en respectant les contraintes liées au projet. En effet ce PAE a été réalisé dans les délais et sans dépassement budgétaire. De plus, nous avons pu établir une solution qui propose à la fois une meilleure expérience utilisateur et qui offre aux développeurs la possibilité de développer dans une architecture robuste, flexible et avec une maintenance facilitée de par sa conception n-tiers.

Avec un peu de recul, la mise en place du PAE a grandement modifié le développement Web au niveau de la société FERCO, en apportant des améliorations de processus (gestion des codes sources, des tests, de mise en production, etc.) et des gains de performance lors du développement.

6.2 Les services développés depuis la mise en service du projet

Le retour positif des utilisateurs et de l'entreprise sur ce projet a permis de lancer d'autres projets Web à intégrer au PAE. Depuis sa mise en ligne, nous avons maintenant traité plusieurs sujets à valeur ajoutée (principalement de la dématérialisation de formulaire avec un workflow associé).

6.2.1 Gestion des visites

Lorsque une personne externe à la société arrive dans l'enceinte de l'établissement, elle est réceptionnée à l'accueil de la société qui enregistre sa venue et contact un collaborateur qui doit accompagner cette personne durant sa visite au sein de l'entreprise, il s'agit là du processus simplifiée de la gestion des visites à FERCO. L'application Web "gestion des visites" implémente ce processus et offre la possibilité aux utilisateurs autorisés de créer des demandes de visite, elles sont ensuite transférées au gestionnaire de l'application qui va devoir effectuer un certain nombre d'actions (réservation d'une salle, d'un hôtel, d'un restaurant, etc.). Un état hebdomadaire de ces visites est envoyé par mail au responsable de la sécurité, aux gardiens ainsi qu'aux responsables de services dans le but de les prévenir que des personnes étrangères de la société sont présentes. Ce projet a pour but de délivrer un journal complet d'entrées/sorties des personnes externes à la société.

6.2.2 Demande de vêtements

L'infirmerie de FERCO doit procéder à la gestion et la livraison des équipements de protection individuelle (EPI). Un EPI est un dispositif destiné à être porté par une personne en vue de la protéger contre un ou plusieurs risques susceptibles de menacer sa santé ainsi que sa sécurité. On retrouve les éléments suivants dans les EPI (liste non exhaustive) : casques, chaussures, bouchons d'oreille, gants, etc. Dans le cadre de cette application, le salarié crée une demande de vêtement qui va ensuite être soumise à son responsable, ce dernier analyse la demande et vérifie si le salarié a dépassé sa dotation autorisée (limite annuelle autorisée par type d'EPI), si le responsable valide la demande, une notification est par la suite envoyée à l'infirmière l'informant qu'une nouvelle demande de vêtement

est initiée dans le système. Suite à cette action, l'infirmière attend l'arrivée du salarié pour procéder à la livraison et mettre à jour les informations dans le système.

6.2.3 Demandes d'investissement

Les demandes d'investissements (DI) peuvent être initiées par tout salarié disposant d'un compte informatique personnel. Ces DI entrent ensuite dans un processus de validation en fonction de différents critères (montant, niveau hiérarchique du demandeur, etc.). Le demandeur est chargé de compléter les données de base de la DI, puis de la soumettre dans le circuit de validation où différents acteurs vont pouvoir approuver ou non la DI. Comme ce processus était effectué à l'aide d'un document papier, il était difficile de suivre l'avancement de la validation des DI en cours (courrier interne, etc.). Un outil de suivi manuel a été instauré au format Excel, complété par le service Achats, afin de suivre les principaux jalons (temps de passage) et la performance Achats (négociations). Au terme de la validation de la DI par la Direction, de l'enregistrement comptable de la DI et de la négociation auprès des fournisseurs, la DI est retournée au demandeur pour qu'il puisse effectuer la création d'une demande d'achat (DA) dans le système SAP, cette étape est obligatoire afin de pouvoir procéder à la commande.

La réalisation de ce sujet à travers une application Web a permis notamment aux utilisateurs de soumettre une DI complète avec des règles métiers qui évitent les erreurs inhérentes du document papier et des inscriptions manuelles, d'obtenir aussi en temps réel l'état d'avancement de leurs demandes et de procéder à des états et extractions automatiques. Ce sujet était le plus complet et le plus complexe à mettre en oeuvre sur le *portail App*, on retrouve en effet l'ensemble des fonctionnalités proposées par le PAE :

— le système de workflow,
— des rapports et indicateurs,
— des formulaires complexes,
— des jobs de traitement dans le but d'envoyer des notifications de relance aux acteurs,
— une interaction avec le système SAP dans le but de récupérer des informations à partir des données de base de la DI,
— des exports PDF de la demande,
— une interface de transmission de pièces jointes, afin de lier des devis et documents à la DI,
— un système de commentaires afin d'enrichir la demande.

6.2.4 Boîte à idées

La boîte à idées est une application Web qui permet à des gestionnaires d'ouvrir des thèmes (réduction de frais généraux par exemple) où les salariés peuvent publier des idées. Les idées sont exportables dans un fichier Excel par le gestionnaire afin d'exploiter les résultats.

6.2.5 PlanetPress

PlanetPress est une application de gestion qui va permettre de piloter les impressions d'une imprimante qui est capable de regrouper les impressions par destinataire et de les mettre sous pli. Cette application est réservée à certaines personnes du service commercial.

6.3 Impact sur l'entreprise

Au départ, ce projet technique ne visait qu'à remplacer une technologie par une autre, cependant la mise en place de ce projet a impacté l'entreprise positivement sur trois aspects : la satisfaction des utilisateurs, les processus des développements informatiques (technologie, méthodologie et documentation) et les coûts des développements.

En effet, la revue des processus des développements informatiques qui a été effectuée par une veille technologique poussée a engendrée une optimisation de la qualité et des performances des développements Web (partie détaillée plus loin dans le document). Cette optimisation a par conséquent permis de réduire les coûts des développements Web de la société qui maintenant dispose d'un PAE qui utilise de manière plus poussée les standards du Web avec une maintenance plus facile et permet un développement plus rapide. Outre la réduction des coûts, comme les utilisateurs sont satisfaits du projet, cela a donné lieu à une résistance au changement très faible lors de sa mise en place et a amélioré le rapport des utilisateurs avec les applications Web développées par l'informatique, cet apaisement permet maintenant d'intégrer plus facilement de nouveaux besoins utilisateurs en leurs offrant un service de qualité.

6.4 Performance et qualité des développements

6.4.1 Les performances du système

Concernant les performances générales du système, comme annoncé dans le cahier des charges sur une page Web basique sans traitement particulier nous attendions un objectif de temps global de chargement de la page Web (transmission des informations du serveur et affichage du rendu par le navigateur) d'une seconde (pour rappel l'Intranet était à environ trois secondes), nous arrivons finalement à un temps global sur une page de test à environ 900ms. Notre objectif est donc atteint et même dépassé ! Ce résultat s'explique par plusieurs facteurs dont notamment le système de cache, ainsi que la gestion des ressources de Symfony 2 (minimiser l'ensemble des ressources en un fichier par exemple pour éviter les multiples requêtes réseaux).

6.4.2 Comparaison Intranet - *portail App*

Aujourd'hui, le développement d'applications Web sur le PAE est devenu bien plus rapide que celui sur l'Intranet. La première raison de ce gain est la technologie utilisée : la conception de Symfony 2 permet un développement bien plus rapide et industrialisé que la conception de Zend Framework 1. SF2 est aussi bien plus sophistiqué, il fournit un ensemble de commandes qui, couplées à celles de Doctrine 2, permet d'automatiser des traitements qui nous auraient pris un grand temps de développement dans l'Intranet. Avec maintenant un retour d'expérience concret sur la création d'applications Web sur le *portail App*, nous estimons le temps de développement sur le *portail App* de deux à cinq fois plus rapide que sur l'Intranet, en fonction de l'application (plus elle sera complexe et de taille importante, plus le temps gagné le sera également). A titre d'exemple, le projet DI qui était un sujet lourd et complexe a été initialement estimé (avec la méthode des points de fonction) à une charge de 2000h sur l'Intranet, il aura finalement été réalisé en 404h de travail sur le *portail App*. Ces observations nous rassurent dans le choix des technologies et des outils qui ont été réalisés.

Nous envisageons maintenant d'abandonner l'Intranet et de procéder à la migration des applications restantes sur le *portail App* sous forme de projet "fil rouge", c'est à dire que les nouveaux projets seront implémentés prioritairement et que la migration des applications se fera en parallèle, ce choix a été justifié par le fait que les deux PAE peuvent cohabiter ensemble sans soucis.

6.5 Perspectives d'évolution du portail d'entreprise

Actuellement, nous n'avons pas de nouveau projet prioritaire à développer, il a donc été décidé de migrer les applications majeures de l'Intranet dans cet ordre de priorité : la gestion des absences, la gestion des plans d'actions et les capacités fonctionnelles. Une fois que ces applications seront réalisées dans le *portail App*, un autre grand chantier sera de procéder à la mise à jour technique de la version actuelle du framework Symfony (2.3) qui utilisée par notre PAE à la nouvelle branche de version (3.X). Comme la version 2.3 est une version Long Term Support (LTS) que SensioLabs s'engage à maintenir quatre ans depuis la date de sortie, soit jusqu'en Mai 2017, nous avons encore du temps pour réaliser ce travail de mise à jour dans le but d'arriver à passer sur la prochaine version LTS de la branche 3.X, à savoir la version 3.4 qui sortira en Novembre 2017.

7 Conclusions

7.1 Déroulement de la mission

Réaliser ce projet a été une expérience enrichissante et complète, comme l'ensemble des tâches ont été confiées à l'équipe interne, nous avions la responsabilité de la globalité du projet depuis la pré-étude en effectuant la veille technologique nécessaire afin de déterminer la meilleure opportunité technique en passant par l'analyse avec les différents acteurs du projet et enfin la réalisation de la solution. Je vais essayer de retracer certains points importants et difficultés survenus lors de ce projet.

Le premier point important était une prise de conscience survenue lors de la phase de pré-étude, nous avions en effet pensé qu'il s'agirait uniquement d'un projet technique, cependant au fur et à mesure de cette pré-étude nous avons mesuré tous les bénéfices et les changements que la mise en place d'un nouveau PAE pouvait apporter dans notre organisation. Une fois cette perspective devenue plus claire, le projet était tout de suite devenu beaucoup plus ambitieux et enthousiasmant : élaborer une nouvelle solution qui va révolutionner le travail des développeurs tout en ne perdant pas de vue la satisfaction de nos utilisateurs en leur apportant une solution ergonomique, robuste et conviviale. C'est dans ce contexte qu'avec mon apprenti Nicolas ALVAREZ nous avions débuté le test et l'étude des différentes technologies candidates pour le projet.

Une des difficultés du projet était l'expression des besoins. L'étude a permis de décrire les spécifications fonctionnelles du projet et celles du sujet de l'application pilote (la gestion des déplacements), c'était une étape importante où l'expression des besoins du sujet pilote a nécessité des réunions avec différents acteurs dont le principal était la responsable fonctionnelle de l'application, malheureusement la détermination de certains points fonctionnels était compliqué et a nécessité plusieurs réunions afin d'être résolus. La gestion de projets au sein de la société FERCO est réalisé à l'aide de la méthodologie de type "Waterfall", c'est à dire où chaque étape importante du projet est validée par un document de référence (voir en annexe). Malgré l'utilisation des documents de références liés à notre méthodologie (cahier des charges, dossier de spécification, etc.), la création de diagrammes UML afin de simplifier la communication avec les acteurs du projet et l'implication forte de ces derniers lors de ces différentes réunions, nous n'avons pas pu éviter des modifications des spécifications lors de la réalisation de l'application. Ceci nous a forcé à adopter une méthode plus agile lors du développement en effectuant régulièrement des points avec les demandeurs pendant la réalisation de l'application. Cette méthode a permis de mettre en évidence certaines incohérences dans les spécifications et a permis également de découvrir des cas imprévus dans les différents scénarios de test. Bien que la gestion de ces imprévus était délicat à gérer (modification ou suppression de codes sources), nous avions été vigilant sur le fait de réaliser une solution la plus modulaire possible à l'aide du développement orienté objet et d'une architecture applicative découplée, de ce fait, ces modifications ont pu être intégrées sans remettre en cause l'intégrité de la solution.

Une autre difficulté était la gestion du temps, en effet outre les modifications des spécifications survenues lors du développement qui pouvaient générer une charge de travail supplémentaire, les

soucis liés à l'apprentissage de ces nouvelles technologies étaient aussi des points qui pouvaient impacter notre planning. Malgré ces aléas, nous avons tout de même réussi à l'aide de ces nouvelles technologies et du suivi du projet (des points rapides étaient réalisés en fin de chaque journée pour donner une dynamique dans le projet) à atteindre les résultats en respectant les délais demandés.

Pendant la réalisation, certains points techniques étaient aussi compliqués à appréhender. Tout d'abord on peut citer la conception de la charte graphique de notre PAE, il fallait en effet livrer une solution ergonomique et conviviale tout en prenant en compte l'aspect de la mobilité. La notion de "responsive design" était inconnue pour mon apprenti et moi même, nous avons donc dû étudier ce concept et l'implémenter en espérant le faire dans les règles de l'art. Ensuite nous avons rencontré des difficultés à implémenter l'authentification automatique avec le protocole NTLM, nous avions d'abord essayé de paramétrer le serveur pour qu'il se charge de cette partie, malheureusement nous avons perdu deux jours sans succès (même si on pensait se rapprocher de la solution), nous avons ensuite décidé d'implémenter le processus d'authentification directement au niveau applicatif ce qui a posé moins de problèmes (moins d'un jour de développement), de plus cette solution ne poserait pas de soucis en cas de mise à jour de l'infrastructure.

7.2 Bilan personnel

Je suis vraiment heureux du choix d'avoir réalisé des cours du soir au CNAM, je pense en effet que ces cours m'ont permis d'évoluer à la fois sur le plan professionnel (compétences techniques, compréhension du monde de l'entreprise et des sciences humaines) et personnel (persévérance, communication et ouverture d'esprit). Je suis maintenant fier d'aboutir à ce mémoire avec la réalisation d'un projet aussi complet qui termine ma formation. Le bouleversement qu'a apporté ce nouveau PAE dans le développement Web de la société m'a clairement montré qu'il faut être très vigilant en informatique pour que ces connaissances techniques ne deviennent pas vite obsolètes tant ce monde évolue à grande vitesse. J'ai aussi beaucoup apprécié réaliser ce projet avec mon apprenti et c'était vraiment intéressant de nous voir évoluer tous les deux techniquement sur le sujet. Même si j'avais la responsabilité de piloter le projet et de prendre les décisions, je tenais à ce qu'elles soient prises le plus possible de manière collaborative afin de créer une réelle dynamique de groupe dans le but de travailler à la fois de façon agréable et efficace. Je suis maintenant déterminé à utiliser non seulement mes connaissances dans le monde professionnel, mais aussi de vouloir les transmettre en espérant ainsi pouvoir contribuer au développement personnel d'autres personnes.

8 Annexes

Annexe A : Statistiques d'utilisation des serveurs Web

Web Servers

Most popular web servers

© W3Techs.com	usage	change since 1 December 2015
1. Apache	55.2%	-0.5%
2. Nginx	27.0%	+0.7%
3. Microsoft-IIS	12.3%	-0.2%
4. LiteSpeed	2.3%	-0.1%
5. Google Servers	1.4%	+0.1%

percentages of sites

Annexe B : Statistiques d'utilisation des langages de programmation

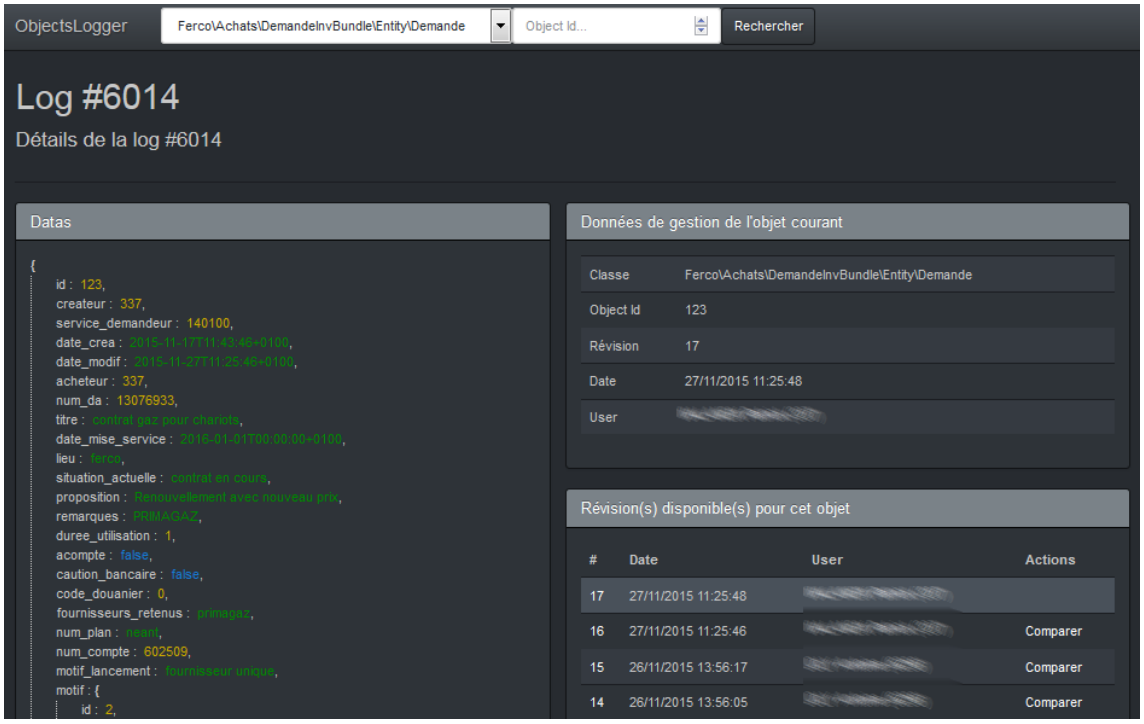
Server-side Programming Languages

Most popular server-side programming languages

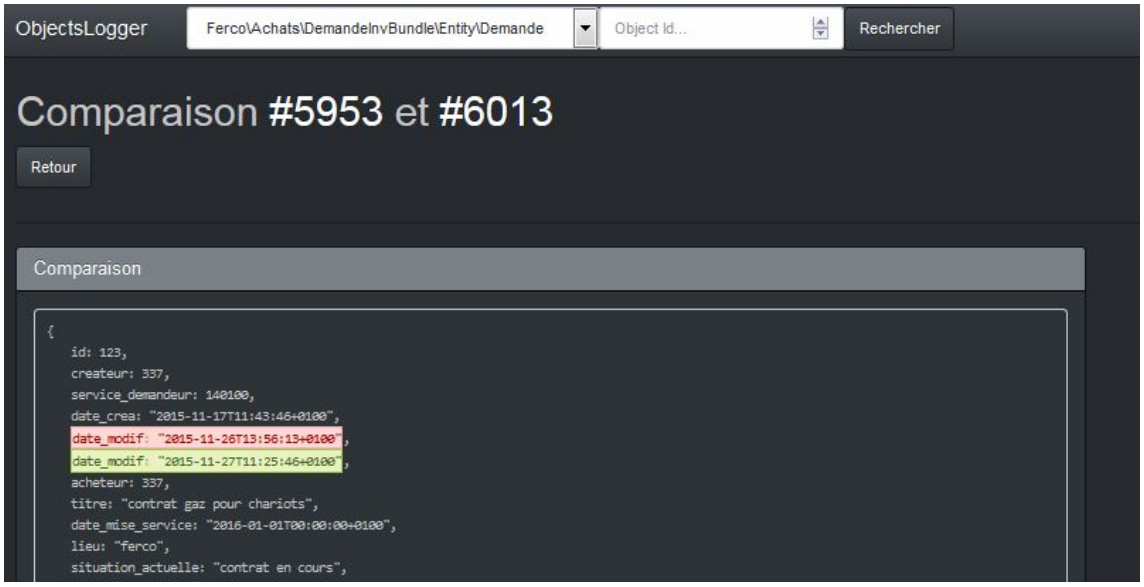
© W3Techs.com	usage	change since 1 December 2015
1. PHP	81.7%	+0.1%
2. ASP.NET	16.0%	-0.1%
3. Java	3.0%	
4. static files	1.6%	
5. ColdFusion	0.7%	

percentages of sites

Annexe C : Interface graphique de l'application *ObjectsLogger*



Annexe D : Comparaison de deux messages de logs dans l'application *ObjectsLogger*



Annexe E : ForceInfo

FERCO

ForceInfo

La gestion des déplacements : un nouvel outil disponible sur intranet !

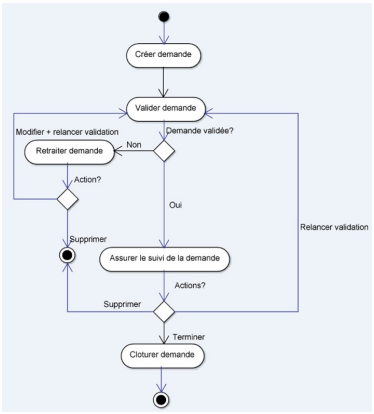
En Bref…

*FERCO organise comme chaque année un **don du sang** au sein de l'entreprise.*

RDV mercredi 11 juin 2014 de 9h00 à 12h00 et de 13h00 à 16h00 en salle G03.

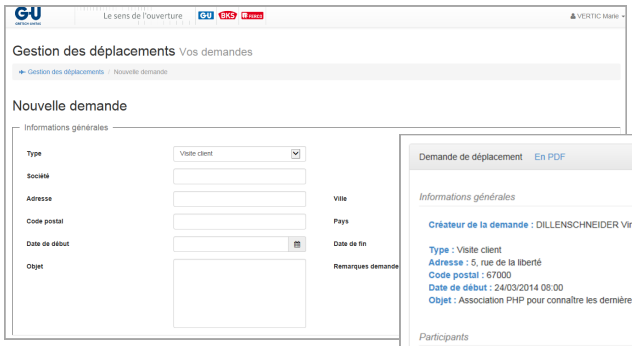
L'objectif de la démarche :

- Remplacer le système fastidieux des feuilles de déplacement
- Mettre en place une gestion des déplacements sur intranet permettant à chacun de gagner en temps et en efficacité



L'intérêt de la démarche :

- Centraliser et standardiser l'ensemble des demandes par le biais d'un outil unique
- Saisir les éléments de façon simple et ludique
- Permettre à chacun et à tout moment de suivre l'évolution de la demande et d'accéder aisément aux réservations
- Permettre aux RH d'avoir des données directement exploitables



Un outil ergonomique, simple d'utilisation, accessible via les applications intranet : en quelques clics, créez vos demandes, suivez leur évolution et accédez aux pièces nécessaires à votre déplacement.

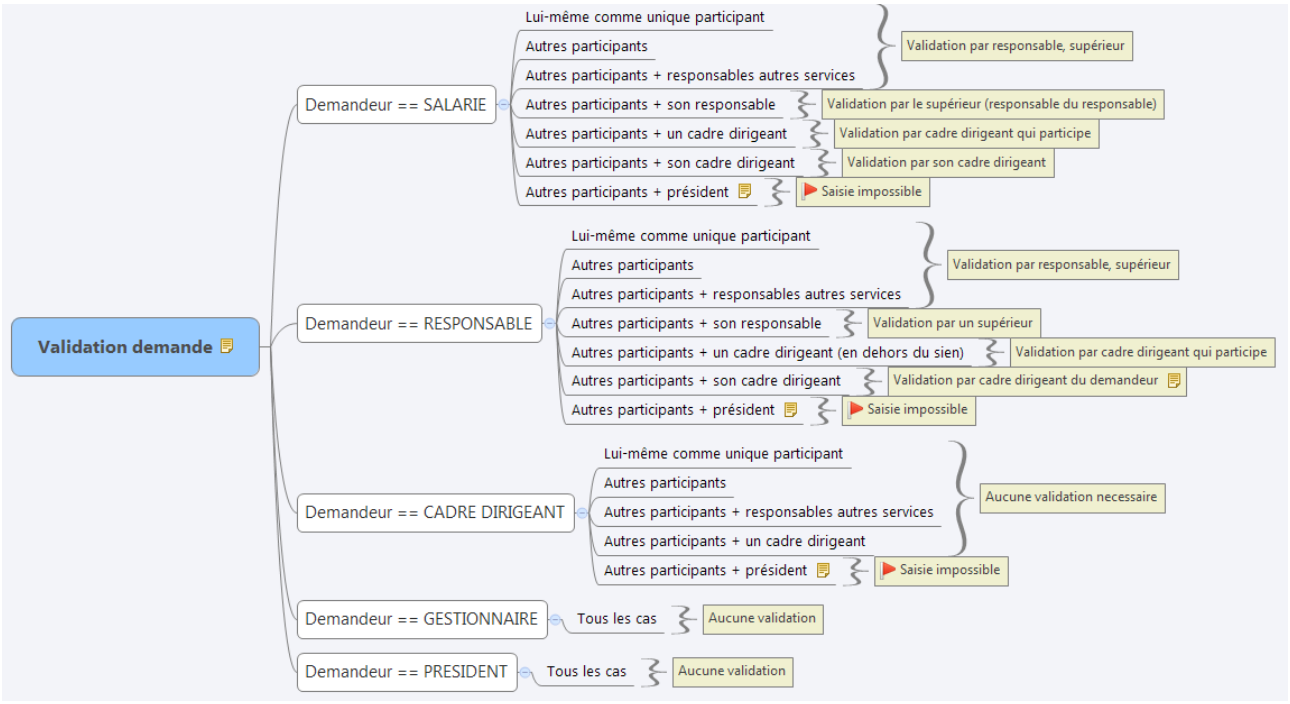
Réalisé avec la collaboration de DILLENSCHNEIDER V. • VERTIC M.

Continuons le progrès pour plus d'efficacité !

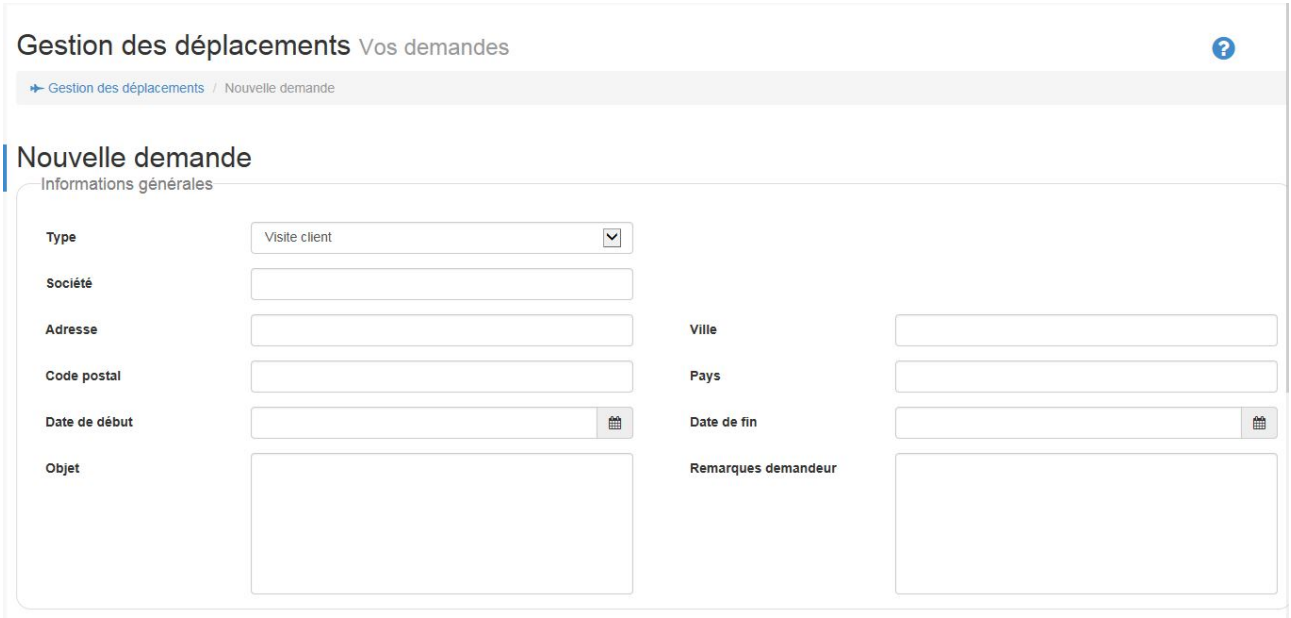
S 24/2014

Le sens de l'ouverture

Annexe F : Schéma de validation processus des déplacements



Annexe G : Exemple de la saisie des informations générales d'un déplacement sur un ordinateur



Annexe H : Exemple de la saisie des informations générales d'un déplacement sur un smartphone

Références

[BAUD, 2012] BAUD, J. (2012). *ITIL - Mise en oeuvre de la démarche ITIL en entreprise, Eni,* ST HERBLAIN.

[GROUSSARD, 2011] GROUSSARD, T. (2011). *Java Enterprise Edition, Eni,* ST HERBLAIN.

[NOGIER, 2008] NOGIER, J. (2008). *Ergonomie du logiciel et design Web, Dunod,* PARIS.

[PAULI, 2009] PAULI, J. , PONCON, G. (2009). *ZendFramework - Bien développer en* PHP*, Eyrolles,* PARIS.

[PORTENEUVE, 2008] PORTENEUVE, C. (2008). *Bien développer pour le Web 2.0, Eyrolles,* PARIS.

[RIMELE, 2011] RIMELE, R. (2011). *HTML5 - Une référence pour le développeur Web, Eyrolles,* PARIS.

[SEGUY, 2007] SEGUY, R., GAMACHE, P. (2007). *Sécurité PHP5 et MySQL, Eyrolles,* PARIS.

[jQuery] Site Internet dédié à la librairie javaScript jQuery. https://jquery.com

[Symfony] Site Internet dédié à la solution Symfony 2. https://symfony.com

[Composer] Site Internet de l'outil de gestion de dépendances pour PHP : Composer. https://getcomposer.org

[html5test] Site qui permet de tester le support et la compatibilité de HTML5 des navigateurs Web. https://html5test.com

Liste des symboles

AJAX L'AJAX (Asynchronous JavaScript and XML) est une architecture informatique qui permet de construire des applications Web interactives.

API Interface de programmation applicative (noté API, en anglais Application Programming Interface) est une interface via laquelle un programme offre un ensemble de services à d'autres logiciels.

ASP.NET ASP.NET est une technique créée par Microsoft afin de générer des pages Web, elle se trouve intégrée dans la plate-forme .NET. ASP.NET est une évolution d'Active Server Pages (ASP). ASP.NET peut être utilisé avec un des langages de programmation de la plate-forme .NET (VB.NET, C#, etc.).

C# C sharp (écrit C#) est un langage de programmation orienté objet destiné à développer sur la plate-forme Microsoft .NET.

CSS Les feuilles de style en cascase (noté CSS, de l'anglais Cascading Style Sheets) est un langage qui permet de mettre en forme les documents HTML et XML.

ECM Le gestionnaire de contenu d'entreprise (en anglais, Enterprise Content Management : ECM) est le terme qui désigne les solutions informatiques qui permettent aux sociétés de gérer leurs contenus numériques.

EDI L'échange de données informatisé (noté EDI) est un terme qui définit un échange d'informations informatique entre deux entreprises ou structures.

ERP Un progiciel de gestion intégré (en angais, Enterprise Resource Planning : ERP), est un logiciel capable de gérer l'ensemble des besoins de l'entreprise à travers une solution d'un concepteur unique.

ExtJs ExtJs est une framework permettant la construction d'applications Web interactives.

framework Un framework est un ensemble d'outils pour le développeur lui permettant de répondre à une problématique.

FTP File Transfer Protocol (FTP) est un protocole de transfert de fichiers qui utilise un réseau TCP/IP.

GIT GIT est le nom d'un logiciel de gestion de versions décentralisée crée par Linus Torvalds.

HTML L'Hypertext Markup Language (noté HTML), est un format de données de présentation pour les pages Web.

HTTP HyperText Transfer Protocol est un protocole de commnication client-serveur développé pour le Web.

IDE L'envrionnement de développement intégré (noté IDE, de l'anglais Integrated Development Environment) est un ensemble d'outils disponible sur la même interface à destination du développeur dans le but de développer plus rapidement et efficacement des logiciels.

JAVA JAVA est un langage de programmation orienté objet. Ce langage est utilisé dans beaucoup de domaines de part sa modularité et sa portabilité sur les systèmes.

JavaScript JavaScript est un langage de programmation principalement utilisé dans les pages Web.

JEE Java Enterprise Edition (noté Java EE, JEE ou encore anciennement J2EE) est une spécification pour JAVA destinée aux applications d'entreprise. Cette spécification préconise le développement d'applications robustes à travers un ensemble de bonnes pratiques et d'API.

JSON JSON (pour JavaScript Object Notation) est un format de données dérivé de la notation des objets du langage JavaScript.

MVC Le modèle vue contrôleur (noté MVC, de l'anglais model view controller) est une architecture dont l'objectif est de séparer les parties d'un programme informatique.

MySQL MySQL est le nom d'une solution de système de gestion de base de données.

NTIC NTIC est la notation abrégée de Nouvelles Technologies de l'Information et de la Communication, ce terme regroupe l'ensemble des technologies liées au Web 2.0.

ORM L'ORM (pour Object Relational Mapping) est une technique de programmation qui consiste à donner la possibilité aux développeurs de travailler « virtuellement » avec une base de données orientée objet à partir d'une base de données relationnelle.

PAE Portail d'applications d'entreprise (abrégé en PAE) désigne une solution qui propose des applications Web afin de fournir des services à travers une plate-forme numérique aux salariés.

PHP PHP : Hypertext Preprocessor est un langage de programmation principalement utilisé pour produire des pages Web dynamiques.

REST REST (pour representational state transfer) est un style d'architecture permettant de construire des applications Web.

SAP SAP désigne le nom d'un ERP propriétaire. Cette solution est délivrée sous forme de modules, on parle par exemple de SAP-HR pour le module lié aux ressources humaines.

SGBD Un système de gestion de base de données (noté SGBD) est un logiciel qui donne la possibilité à un programme d'enregistrer et de partager ses données persistantes au sein d'une base de données.

SI Le système d'information (noté SI) est l'ensemble des ressources disponibles permettant de traiter l'information.

SOAP SOAP est l'acronyme de Simple Object Access Protocol qui est un protocole de communication entre deux programmes et qui permet la transmission de messages entre deux objets distants.

SVN Subversion (en abgrégé SVN) est le nom d'un logiciel de gestion de versions. Il s'appuie sur le principe du dépôt centralisé et unique.

Symfony 2 Symfony 2 est un framework Web MVC lancé par la société SensioLabs en 2011 qui a été développé avec le langage de programmation PHP.

Twig Twig est le nom d'une solution de moteur de template pour le langage PHP.

XML XML est un langage de structuration de l'information qui permet de représenter une information sous forme de balises.

Zend Framework Zend Framework est un framework Web MVC lancé par la société Zend depuis 2006, ce framewok a été écrit dans le langage de programmation PHP.

Table des figures

Résumé

La communication et la gestion de l'information sont des éléments clés au sein d'une société. Un portail d'application d'entreprise (PAE) apporte à travers la technologie Web une solution adaptée aux salariés qui disposent d'un accès personnalisé à l'ensemble des services que propose l'entreprise. La société FERCO a mis en place son premier PAE en 2008, aujourd'hui ce PAE atteint un stade où la maintenance devient difficile en raison des anciennes technologies utilisées lors du développement. De plus, les salariés de la société expriment également de nouveaux besoins, qui sont plus complexes à implémenter dans ce portail applicatif. C'est pourquoi, il est maintenant nécessaire d'étudier et de mettre en oeuvre un nouveau PAE en utilisant les dernières technologies du marché.

Ce mémoire d'ingénieur consiste à découvrir les grandes étapes du projet d'élaboration d'un nouveau PAE en utilisant les nouvelles technologies Web.

Mots clés : portail d'application d'entreprise, FERCO, technologies Web.

Summary

The communication and information management are keys elements in a society. An enterprise application portal (EAP) with Web technology brings the better solution to the employees who can use a personal access to the entire services offered by the company. The FERCO enterprise released his first EAP in 2008, today this EAP has some maintenability issues explained by the old technologies which have been used during the developement. In addition, the employees ask now for more features which are more complicated and difficult to implement in the current EAP. That's why, It is now necessary to analyse and deploy a new EAP with the latest Web technologies.

This engineering thesis presents the steps of deploying the new EAP's project by using the latest Web technologies.

Key words : enterprise application portal, FERCO, Web technologies.