

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement 2016-2017

Réseau d'entreprise : étude et mise en œuvre pour l'équipe RFAI

Tuteur académique
Nicolas RAGOT

Étudiant
Mathieu BOUTOLLEAU (DI5)

3 avril 2017



Liste des intervenants

Nom	Email	Qualité
Mathieu BOUTOLLEAU	mathieu.boutolleau@etu.univ-tours.fr	Étudiant DI5
Nicolas RAGOT	nicolas.ragot@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Mathieu Boutolleau susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Nicolas Ragot susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Mathieu Boutolleau, *Réseau d'entreprise : étude et mise en œuvre pour l'équipe RFAI*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2016-2017.

```
@mastersthesis{
  author={Boutolleau, Mathieu},
  title={Réseau d'entreprise : étude et mise en œuvre pour l'équipe RFAI},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2016-2017}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	v
Introduction	1
I Outils de collaboration pour l'équipe RFAI	2
1 Contexte de la réalisation	3
1 Acteurs	3
2 Contexte de la réalisation	3
3 Objectifs	3
4 Hypothèses	4
2 Description générale	5
1 Environnement du projet	5
2 Caractéristiques des utilisateurs	6
3 Fonctionnalités du système	7
4 Architecture générale du système	8
3 État de l'art	11
1 Création d'un site collaboratif avec Wordpress	11

1.1	Introduction	11
1.2	Fonctionnalités générales	12
1.3	Fonctionnalités en détails	12
1.4	Plugins	15
2	Gestion de projet avec un site Wordpress	16
2.1	Introduction	16
2.2	Plugins Gratuits	16
2.3	Plugins Payants	18
2.4	Résumé	19
3	Collaboration en entreprise avec ONLYOFFICE	19
3.1	Présentation générale	19
3.2	Installation	20
3.3	Sécurité	21
3.4	Intégration	21
3.5	Présentation des fonctionnalités	22
3.6	Présentation des outils	25
4	Listes de diffusion pour l'équipe RFAI	26
4.1	Flux RSS avec Wordpress	26
4.2	Listes avec ONLYOFFICE	27
4.3	Listes de l'université	27
4.4	Sympa	28
4	Analyse et conception	30
II	Nouvelles fonctionnalités sur Pilote	31
5	Contexte de la réalisation	32
1	Objectifs	32
2	Hypothèses	33
3	Bases méthodologiques	33
6	Description générale	36
1	Environnement du projet	36
2	Caractéristiques des utilisateurs	36
3	Fonctionnalités du système	36
4	Structure générale du système	37
7	Analyse et conception	39

8	Mise en œuvre	40
1	Architecture existante	40
2	Organisation mise en place	41
3	Résultats.....	41
4	Gestion des versions	42
III	Bilan	45
9	Travail accompli	46
10	Qualité	47
11	Perspectives	50
	Conclusion	51
	Annexes	52
IV	Outils de collaboration pour l'équipe RFAI	53
A	Description des interfaces externes du logiciel	54
B	Spécifications fonctionnelles	57
C	Spécifications non fonctionnelles	60
V	Nouvelles fonctionnalités sur Pilote	62
D	Description des interfaces externes du logiciel	63
E	Spécifications fonctionnelles	65
F	Spécifications non fonctionnelles	67
G	Gestion de Projet	69
H	Documentation livrée	70
1	Présentation de Pilote.....	70
2	Manuel utilisateur	71
3	Manuel d'installation	74
4	Manuel de maintenance	78
5	Manuel du contribution	78

Webographie	81
Bibliographie	82

Table des figures

2 Description générale

1	Frontière entre le site du LI et le wiki RFAI.....	6
2	Cas d'utilisation par les responsables.....	7
3	Cas d'utilisation par l'équipe RFAI.....	8
4	Architecture générale des outils de collaboration	10

3 État de l'art

1	Exemple de relevé d'activité.....	13
2	Page de présentation d'un groupe.....	14
3	Liste de tâches avec Kanban for Wordpress.....	17
4	Liste des tâches d'un projet avec Orbis et Orbis Tasks.....	17
5	Page d'accueil du portail web d'ONLYOFFICE.....	21
6	L'édition de document, en ligne, avec un navigateur	22
7	Exemple de diagramme de Gantt d'un projet.....	23
8	Exemple de calendrier personnel.....	25
9	Flux RSS des posts étiquetés Events	27
10	Création d'une liste de diffusion (image du centre d'aide d'ONLYOFFICE)	28
11	Index des listes de diffusion de l'université	29

5 Contexte de la réalisation

1	Trois écrans principaux de Pilote.....	35
---	--	----

6 Description générale

1	Diagramme des cas d'utilisations de Pilote	37
---	--	----

2	Architecture générale de Pilote	38
8	Mise en œuvre	
1	Les différents modules (bundles) de Pilote, repris du rapport de PFE de Rémi Patrizio	40
2	Diagramme de classe du contrôleur du tableau de bord	42
3	Diagramme de classe de la vue du tableau de bord	43
4	Interface réalisée, en mode réplée, pour le tableau de bord	43
5	Interface réalisée, en mode dépliée, pour le tableau de bord	44
10	Qualité	
1	Documentation du contrôleur du tableau de bord, crée avec PHPDoc	48
2	Tableau de bord d'un projet vide	49
3	Analyse par SonarQube du code PHP de Pilote	49
4	Analyse par SonarQube du code du contrôleur de l'avancement du projet	49
A	Description des interfaces externes du logiciel	
1	Maquette d'interfaces	56
B	Spécifications fonctionnelles	
1	Calendrier d'équipe, publiquement affiché sur le wiki RFAI	58
D	Description des interfaces externes du logiciel	
1	Tableau Kanban d'un projet	63
E	Spécifications fonctionnelles	
1	Maquette de l'IHM du tableau de bord projet	66



Introduction

Ce document est le rapport final de mon Projet Recherche & Développement (PRD) portant sur la mise en place d'un réseau social d'entreprise à destination de l'équipe Reconnaissance des Formes et Analyse d'Images (RFAI) du Laboratoire d'Informatique (LI) de l'Université de Tours. Ce projet s'est déroulé tout au long de l'année scolaire 2016-2017.

Ce rapport présente la problématique du projet et les solutions explorées. La solution proposée repose sur une réflexion particulière qui vise à proposer à l'équipe RFAI les outils pour mieux collaborer au quotidien sans bousculer ses habitudes de travail. Le PRD s'est déroulé en deux phases, quasiment indépendante l'une de l'autre.

La première partie du projet ne comportait pas de développement et elle s'est concentré autour de réflexions et de la rédaction de plusieurs documents présentant différents outils pour améliorer la collaboration des membres de l'équipe RFAI.

Quant à la deuxième partie, présentée dans la **Partie II**, elle ne comportait pas de partie dédiées à l'analyse et à l'étude. Elle était dédiée au développement de nouvelles fonctionnalités sur un logiciel existant.

Une dernière partie s'occupera de faire le bilan de ce projet, en présentant le travail accompli et en s'attardant sur la qualité du code produit.

Ce rapport est une version revue et augmentée du rapport qui a été rendu au semestre 9. Il comporte également de nombreux documents annexes dont la présence est nécessaire. Tout cela en fait un rapport assez imposant, néanmoins je vous souhaite une bonne lecture.

Première partie

Outils de collabora-
tion pour l'équipe RFAI

1

Contexte de la réalisation

1 Acteurs

Cette partie du document est une présentation d'outils de collaboration à destination de l'équipe RFAI du LI. L'équipe RFAI souhaite mettre en place des outils lui permettant de mieux se faire connaître par le public et de pouvoir faire travailler efficacement ses membres.

M. Nicolas Ragot, maître de conférences au LI et coresponsable de l'équipe RFAI avec M. J-Y. Ramel, a proposé un Projet de Recherche et Développement (PRD) afin d'installer un tel outil. Il représente la maîtrise d'ouvrage (MOA). Elle est aussi composée des responsables techniques de Polytech' Tours, chargé du déploiement et de la maintenance de l'outil, MM. P. Meichel et M. Rousseau.

L'auteur de ce document est Mathieu Boutolleau, étudiant en 5e année et chargé de la réalisation de ce PRD, qui représente la maîtrise d'œuvre (MOE).

2 Contexte de la réalisation

Le projet s'inscrit dans une volonté de modernisation du site web de l'équipe RFAI. L'équipe a voulu améliorer sa vitrine sur internet qui lui permet de se présenter au public et ajouter un outil de collaboration pour ses membres.

3 Objectifs

Il existe déjà un site web de l'équipe RFAI à destination du public et géré par celle-ci. Il a vocation d'être un complément au site officiel de l'équipe, on le retrouvera ici : [[WWW9](#)].

Ce site web, permet à l'équipe RFAI de publier des informations et de mettre à dispositions des données scientifiques, comme des benchmarks et des prototypes. Ces publications ne sont pas actuellement possibles sur le site du LI et il n'est pas souhaité de passer par une plate-forme d'hébergement externe. Il comprend deux parties : une publique et une privée. On peut le consulter en suivant ce lien : [[WWW4](#)].

La partie publique, accessible à tous par internet, est un site web présentant l'équipe, des actualités, ses activités de recherche et ses travaux en cours. Le site est sous forme de wiki pour

que chaque membre puisse éditer des pages du site. Cette partie a pour vocation à être la vitrine de l'équipe sur internet.

La partie privée sera uniquement accessible aux membres de l'équipe RFAI, par une étape d'authentification obligatoire. Elle possède deux objectifs principaux :

- améliorer la communication et la collaboration au sein de l'équipe ;
- centraliser et partager des documents ainsi que de l'information.

La solution mise en place sera déployée sur des machines mises à disposition par le service informatique de l'école, l'objectif est que les personnes en charges de la maintenance gardent une maîtrise de l'endroit et des serveurs utilisés par l'outil collaboratif.

4 Hypothèses

La mise en place site web collaboratif s'est accompagné d'une phase de recherche sur l'état de l'art. Il a fallu voir ce qu'il était possible de faire avec des solutions existantes, l'architecture du système d'information de l'université et les règles de sécurité en vigueur.

La politique informatique de l'université précise les pré-requis à remplir afin de pouvoir déployer un site web public. Elle restreint les solutions et les méthodes qu'il est possible d'employer pour ce projet.

Le site public actuel a été conçu à l'aide d'une charte de responsabilité, cette solution est conforme à la charte en vigueur et est jugée viable. En revanche toute modification de technologies utilisées ou l'ajout de greffons au site présente un risque important de ne pas être accepté. Dans le cas de l'impossibilité de mettre en place un greffon ou une technologie à cause de cette charte, on cherchera à proposer une alternative de remplacement ou de développer notre propre solution.

L'école d'ingénieur possède un annuaire LDAP listant les différents utilisateurs du parc informatique et leurs droits. Il est souhaité de les retrouver dans notre solution, cela permettra de ne pas avoir à y redéfinir ces éléments. En effet, on aura alors le fonctionnement habituel des outils de l'école avec l'équipe technique qui sera chargée de définir et maintenir les accès aux outils. On cherchera à intégrer notre outil à cet annuaire, si cela n'est pas possible, il faudra envisager de créer une passerelle ou de gérer les droits d'accès directement depuis l'application.

2

Description générale

1 Environnement du projet

La solution s'intégrera aux sites web déjà en place et hébergés au sein de l'Université de Tours. Notamment elle cohabitera avec le wiki déjà existant de l'équipe RFAI. Ce wiki est déployé sur un espace d'hébergement mutualisé mis à disposition par l'Université.

Comme nous l'avons abordé dans la [Section 4](#) (Chapitre 1), l'université a mis en place une politique qui régit le déploiement de site web au sein de l'université. Elle oblige ainsi tout site web mis en place à respecter des contraintes strictes. Elle a été signée par M. M. Rousseau et impose des engagements pris à la fois en termes d'évolution et de maintenance, mais aussi en termes de sécurité.

Notre solution sera déployée sur des serveurs de l'université, sa maintenance devra être facilement réalisable par des membres de l'équipe technique, principalement par ceux rattachés à l'école. Le wiki de l'équipe RFAI est accessible au public via Internet, il est obligatoire de mettre en place des mesures adéquates de sécurité afin de garantir l'intégrité des machines et du réseau de l'université. Ces deux critères sont extrêmement importants, notre solution devra être conformes aux différents pré-requis soulevés.

Concernant la situation actuelle, nous avons rapidement présenté les pages de l'équipe RFAI sur le site actuel du LI dans la [Section 3](#) (Chapitre 1). Ce site est la page officielle de l'équipe et est conçu avec K-Sup (on retrouvera plus d'informations sur le site officiel, [\[WWW6\]](#)), un système de gestion de contenu libre (licence Apache, version 2.0 [\[WWW1\]](#)) à destination des institutions de l'enseignement supérieur. Une page d'accueil présente le LI avec un mot du directeur, M. Jean-Charles Billaut. On retrouve aussi les offres de postes et de thèses en cours ainsi que les dernières actualités du laboratoire. Ensuite des rubriques dédiées permettent de mettre en avant les trois équipes, les formations dispensées, les publications et les projets du LI.

Le site du LI ne permet pas de publier des pages professionnelles pour les membres du laboratoire, présentant par exemple leurs recherches et enseignements. Ces pages sont donc rattachées au wiki de l'équipe RFAI. Il existe un espace commun où chaque membre de l'équipe peut déposer une page web personnelle. Initialement, le wiki était un site sous SharePoint (solution de Microsoft pour créer des sites et portails web), aujourd'hui il a été remplacé par une version faite avec le système de gestion de contenu libre WordPress (sous licence GPLv2 [\[WWW7\]](#), plus d'informations sur le site officiel [\[WWW10\]](#)). Néanmoins un effort particulier a été apporté

afin de garder les mêmes liens vers les différentes pages entre l'ancienne et la nouvelle version du wiki. Ainsi, les liens vers les outils, démonstrations, etc. publiés précédemment dans des articles sont toujours valides.

Le wiki sert à l'équipe RFAI à présenter des éléments complémentaires au site du LI. On retrouve la liste des thèses en cours ou passées et des celle habilitations à diriger des recherches effectuées. Les différents projets en cours ou réalisés ainsi qu'une page vitrine pour des outils et démonstrations réalisées par des membres de l'équipe. Une dernière partie propose des ressources, des actualités et événements concernant l'équipe RFAI également des offres de poste, stage ou thèse. Les événements sont ajoutés par les membres de l'équipe RFAI, un calendrier permet à tous de les consulter.

La **Figure 1** schématise la relation entre le public et l'équipe RFAI et les deux sites actuellement en ligne : celui du LI et le wiki de l'équipe RFAI.

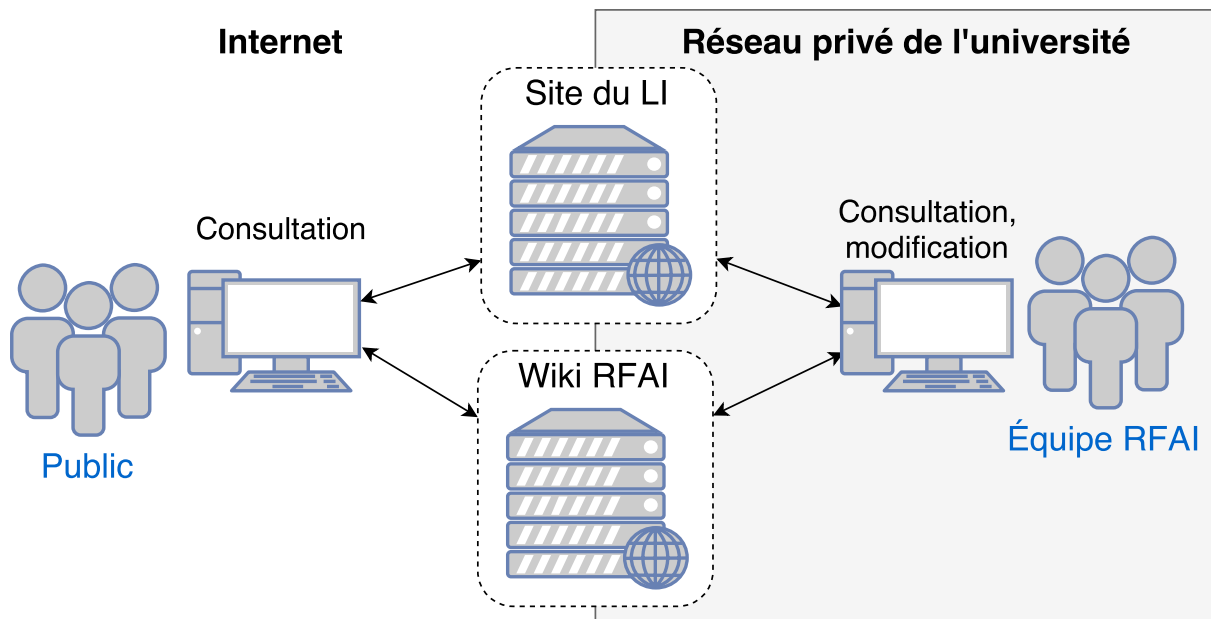


Figure 1 – Frontière entre le site du LI et le wiki RFAI

2 Caractéristiques des utilisateurs

Il y a trois types d'utilisateurs du site web collaboratif :

- Le public : utilisateurs occasionnels ou régulier, ils peuvent uniquement accéder à la partie publique ;
- Les membres de l'équipe RFAI, soit 39 personnes en 2016 (dont 20 permanents), ils ont accès à la partie publique et privée du site web. Ils ont une connaissance en informatique, une bonne expérience de l'outil et sont chargés de produire le contenu mis en valeur par la solution. La partie privée sera composée de la partie permettant de modifier le wiki et de celle de gestion des projets de l'équipe ;
- Les responsables du contenu, il s'agit pour le moment des deux responsables de l'équipe RFAI (MM. N. Ragot et J-Y. Ramel). Ils sont chargés du contrôle et de la supervision du contenu publié sur le site.
- Le responsable technique, pour l'instant M. M. Rousseau. Il a accès à tous les éléments du site, y compris à la machine qui l'héberge. Il est responsable de la maintenance générale du site web.

3 Fonctionnalités du système

Le site doit pouvoir être consulté par le public via Internet. Après une étape d'authentification, les membres de l'équipe RFAI peuvent publier ou modifier du contenu sur le wiki au travers d'une interface dédiée. L'équipe a également accès aux outils de collaboration disponibles depuis le réseau local de l'école : l'agenda, les mails et les documents partagés.

La **Figure 3** représente le diagramme général des cas d'utilisation de la solution par le public et l'équipe RFAI.

Pour superviser l'utilisation et le fonctionnement de l'outil, le responsable du contenu pourra veiller sur les éléments publiés sur le wiki et le responsable technique va s'occuper de sa maintenance et de son bon fonctionnement.

La **Figure 2** représente les cas d'utilisation des responsables du contenu et technique.

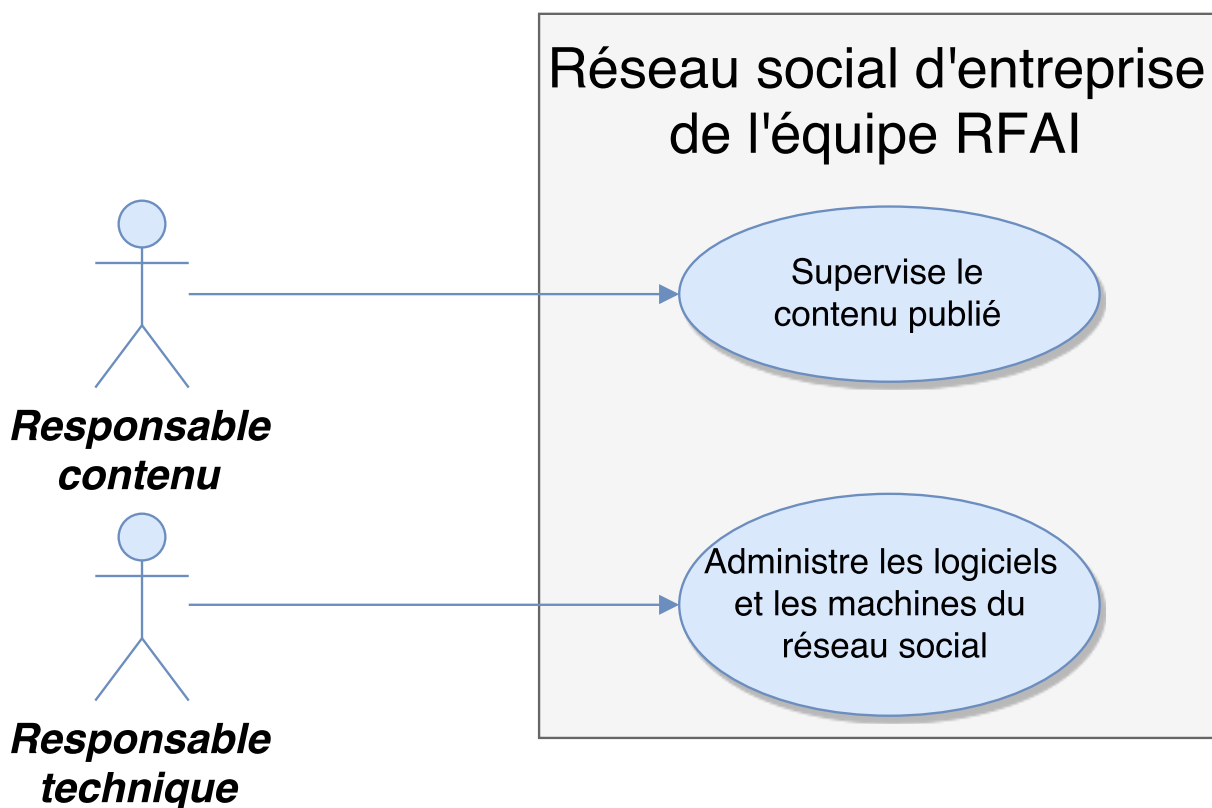


Figure 2 – Diagramme des cas d'utilisation de la solution par les responsables

L'outil de collaboration est uniquement accessible aux membres de l'équipe RFAI et aux personnes chargées de la supervision à partir du réseau privé de l'université. Néanmoins, il est souhaité qu'un accès soit possible depuis un poste à l'extérieur, par exemple lors d'un déplacement professionnel. Pour cela l'école dispose d'un service de VPN (Virtual Private Network, réseau privé virtuel) permettant d'atteindre le réseau local depuis n'importe quel poste relié à Internet. L'accès à notre solution sera disponible via ce service de VPN.

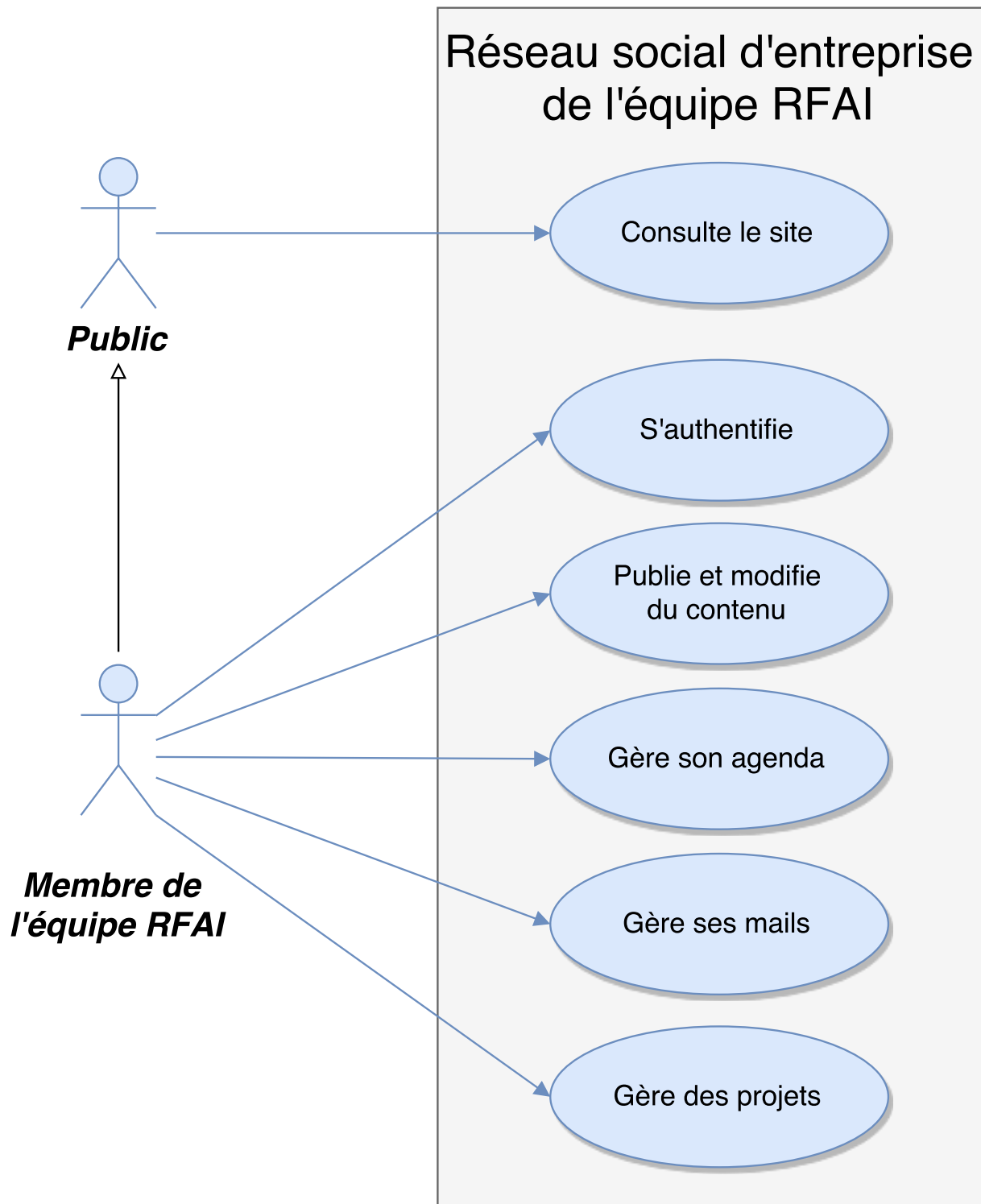


Figure 3 – Diagramme des cas d'utilisation de la solution par l'équipe RFAI

4 Architecture générale du système

Les outils de collaboration proposés sont là pour apporter des solutions pour que l'équipe RFAI puisse travailler efficacement. Les deux principaux points à améliorer, comme présenté dans la [Section 3](#) (Chapitre 1), sont la communication inter-équipe en apportant une nouvelle gestion

des mails. Ainsi que le partage de documents en proposant un stockage commun de fichiers pour l'équipe et un outil pour modifier des documents de manière collaborative.

Ainsi la solution est composée de plusieurs outils permettant de répondre à chacun des besoins de l'équipe RFAI.

Tout d'abord la problématique principale en termes de communications est que les membres de l'équipe (et principalement les responsables) reçoivent un nombre important de mails concernant des communications diverses. Ces éléments sont souvent intéressants à communiquer à toute l'équipe, mais cela soulève un souci de diffusion continue de mails d'informations superflus.

Pour y remédier, nous testerons la mise en place d'une solution de liste de diffusions où chaque membre pourra déposer un mail à diffuser à l'équipe et le classer dans des catégories pour identifier son contenu. Ensuite, les membres de l'équipe RFAI s'abonneront aux listes qui les intéressent et recevront une communication choisie.

L'autre souhait de l'équipe RFAI est d'améliorer sa façon de travailler en coopération. L'accent est mis sur le partage et l'édition commune de documents. Ces documents sont, par exemple, créés lorsque des membres de l'équipe travaillent sur un projet. Il s'agit de documents textes, de classeurs et de présentation au format Office Open XML (standard créé par Microsoft, utilisé par défaut par Microsoft Office depuis sa version 2007).

Une première solution sera de proposer un espace de stockage sur le réseau de l'école pour y déposer ces documents. Ensuite, l'intégration d'un autre outil permettra de mettre en place un portail web pour accéder à ces fichiers et pouvoir les éditer en ligne, éventuellement à plusieurs.

La **Figure 4** présente l'architecture globale de la solution et listant les principaux modules et par quels moyens il sera possible d'y accéder.

Outils de collaboration

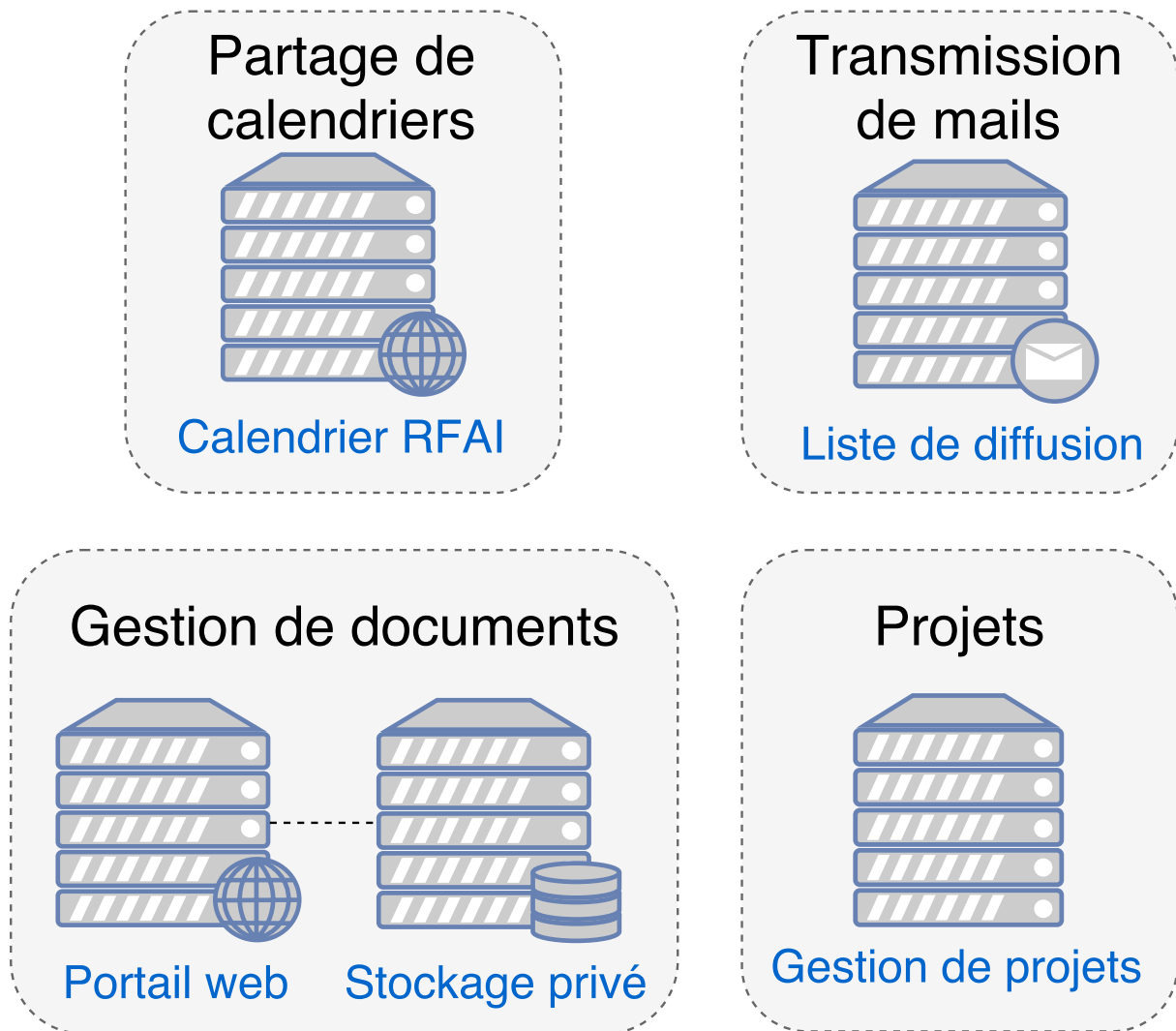


Figure 4 – Architecture générale des outils de collaboration

3

État de l'art

1 Création d'un site collaboratif avec Wordpress

1.1 Introduction

Wordpress

Wordpress est un CMS gratuit et open-source se basant sur PHP et MySQL. La première version date de mai 2003. Il permet facilement de créer son blog et de publier son contenu. Un système de plugins permet d'étendre ses fonctionnalités.

BuddyPress

C'est un plugin de Wordpress, le site officiel est listé ici [[WWW2](#)], permettant de transformer un simple blog en un vrai réseau social. Il est devenu un plugin officiel en 2008 lorsque son développeur a été engagé par la société derrière Wordpress. Une version 1.0 est sortie un an plus tard.

BuddyPress est aussi personnalisable avec des thèmes et plus de 700 plugins dédiés. Il est actuellement déployé sur plus de 200 000 sites.

Commons In A Box (CBOX)

Il s'agit d'un ensemble de plugins ainsi que d'un thème compatible avec BuddyPress (CBOX est donc dédié à la création de sites communautaire). On retrouvera plus d'information sur le site officiel [[WWW3](#)]. C'est un projet de la CUNY Academic Commons, qui vise à effacer la difficulté de création d'un site web communautaire basé sur BuddyPress.

Pour tester la création d'un réseau social avec Wordpress, j'ai installé BuddyPress ainsi que plusieurs plugins (la plupart installé par CBOX pour enrichir les fonctionnalités de BuddyPress). CBOX n'étant pas à jour avec les dernières versions de BuddyPress et de quelques plugins, il ne

permet pas de tester tout ce que je voulais. Il n'est donc pas utilisé pour les tests (mais permet de déployer rapidement un réseau social) car il manque principalement le partage de fichier.

1.2 Fonctionnalités générales

Voici les fonctionnalités offertes par différents plugins qui pourraient intéresser l'équipe RFAI. Tout d'abord, BuddyPress permet de se créer un réseau personnel avec membres, activité, notifications, profil, amis et messages. Pour enrichir notre réseau et en développer d'autres dans notre structure, nous avons les groupes, toujours avec BuddyPress. Il permet créer des groupes d'utilisateurs autour d'un thème de recherche commun, d'une équipe de travail, etc.

On retrouve le travail en collaboration avec BuddyDrive qui permet de partager des fichiers. Il est possible de les partager publiquement, avec un groupe ou avec des amis (test impossible pour le moment). Dans le même registre, l'édition de documents textes avec BuddyPress Docs permet l'édition collaborative de documents avec historique, pièces-jointes et labels.

Enfin, on peut envisager de déployer une wiki, grâce au greffon BuddyPress Docs Wiki. Ce wiki sera en place sur tout le site, il se base sur le module d'édition de documents textes.

1.3 Fonctionnalités en détails

Membres

Les membres seront la base de tout le contenu du site, la création se fait selon deux méthodes. Soit on destine le site à être assez privé et donc les nouveaux utilisateurs seront ajoutés manuellement par un administrateur. Soit on met une page d'inscription à disposition, les inscriptions peuvent être validées ou non par un administrateur.

On peut restreindre l'accès du site aux personnes authentifiées afin de, par exemple, séparer le site en deux :

- une partie publique avec un blog, des documents en accès libre...;
- une partie privée où les membres peuvent échanger et collaborer.

Activité

BuddyPress est un réseau social, donc les différents membres vont faire vivre le site. Une page Activity (voir [Figure 1](#)) présente les derniers événements survenus, comme un changement de statut, la modification d'un document, la création d'un groupe public, etc.

Profil

Permet de créer des profils de membres qui reflètent la communauté présente sur le site. Il est possible de définir les types de champs à renseigner et de définir ceux obligatoires. Chaque membre peut renseigner ce qu'il veut et définir ce qui sera visible aux autres.

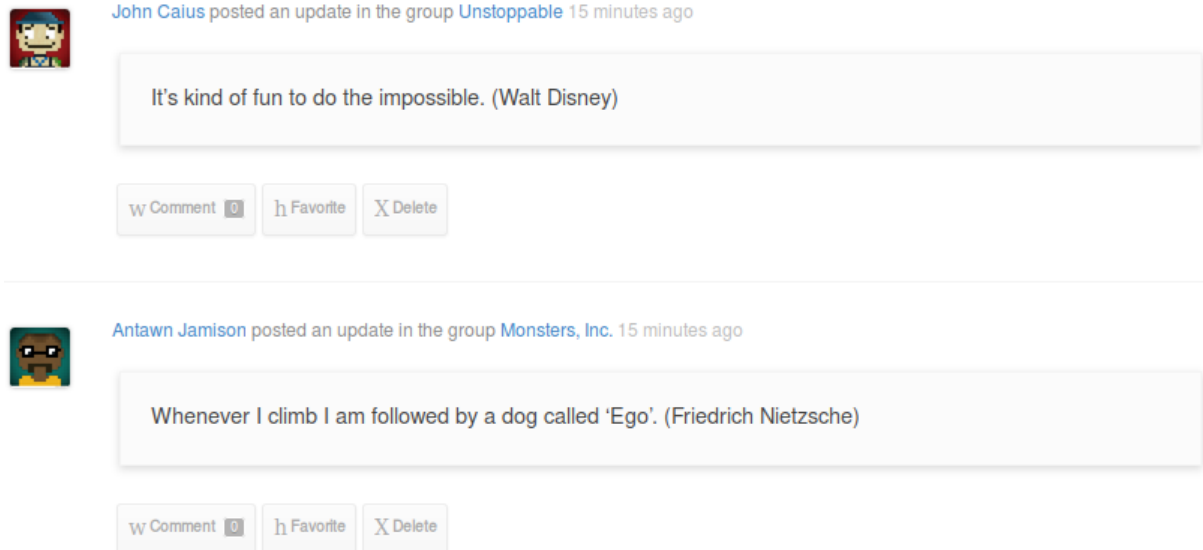


Figure 1 – Exemple de relevé d'activité

Notifications

Il existe deux manières de notifier un utilisateur de l'activité sur le site, soit depuis le site, soit par un e-mail.

Les notifications sur le site apparaissent dès que :

- notre nom est mentionné (eg. @username);
- un membre répond à un de nos commentaires;
- on nous a envoyé un message;
- un membre nous ajoute/accepte en ami;
- il y a de l'activité dans un de nos groupes.

Une page de paramètres permet de régler finement les e-mails que l'on veut recevoir.

Messages

BuddyPress supporte les conversations entre membre(s) avec affichage de notifications lors de l'arrivée d'un message. Cela se rapproche plus du mail que de la messagerie instantanée.

Amis

Un membre peut demander à un autre de devenir amis afin de faire ressortir les membres avec qui il est proche. Cette relation est publique.

Groupes

Les groupes (page Groups) permettent à plusieurs personnes de communiquer efficacement autour d'un même thème, on retrouve un exemple dans la [Figure 2](#).

Les membres peuvent poster des statuts, partager des fichiers avec le groupe, écrire des documents de manière collective. Les notifications d'activité dans le groupe sont personnalisables

par chaque membre (recevoir par mail, différents intervalles possibles, les modifications ou non).

On peut définir certains membres d'un groupe comme administrateurs ou modérateurs, ils seront chargés de la gestion du groupe. On peut aussi restreindre facilement l'accès à certaines fonctionnalités (comme la création/édition de documents du groupe) à certains rôles.

Un groupe peut être public (les membres et le contenu est public, tout le monde peut rejoindre), privé (le contenu est accessible uniquement aux membres, le groupe apparaît dans les résultats, les membres inscrits du site peuvent demander à le rejoindre) ou caché (le contenu est accessible uniquement aux membres, il n'apparaît pas dans les recherches, on peut le rejoindre seulement en y étant invité).

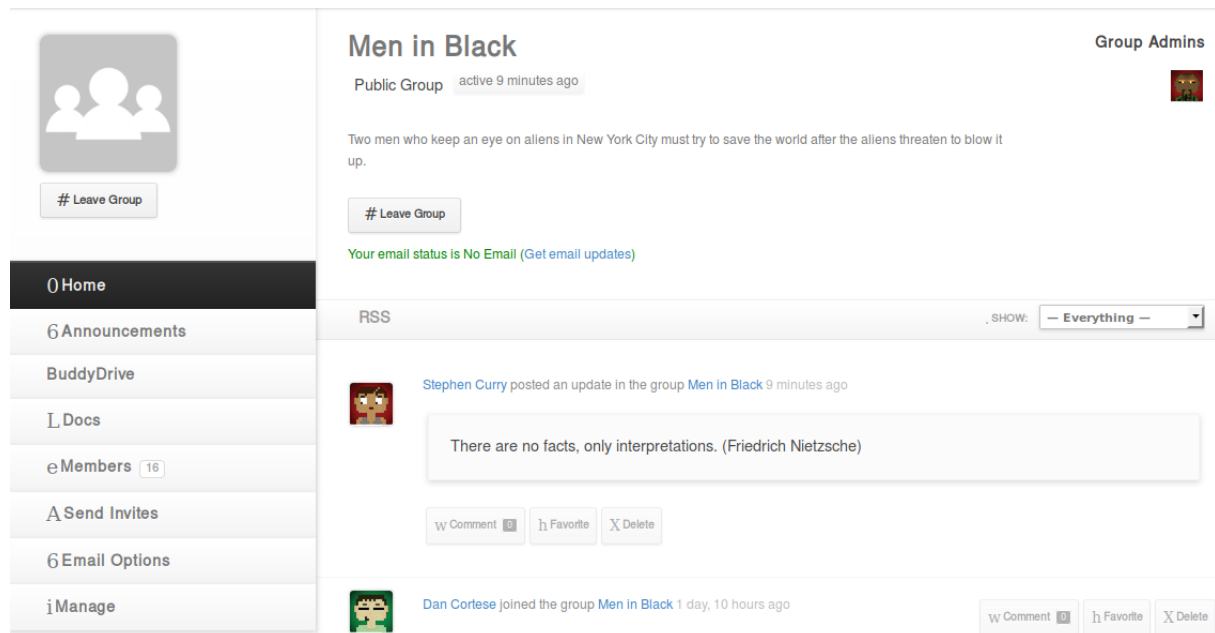


Figure 2 – Page de présentation d'un groupe

Édition de documents

On a aussi la possibilité de modifier des documents textes de façon collaborative (ou non). Les documents peuvent être liés à un groupe, annotés et commentés.

Un historique des changements est gardé, on peut aussi ajouter des pièces-jointes ou des étiquettes pour classer le document.

Le module d'édition de documents et le module de partage de fichier ne peuvent pas gérer les mêmes fichiers.

Wiki(s)

On peut créer deux types de wiki, un public et un privé. Un wiki privé existe à l'échelle d'un groupe, le public pour tout le site.

Pour chaque wiki on peut définir des étiquettes et une hiérarchie sur les pages pour les organiser. Chaque modification et son auteur est tracée et on peut facilement revenir en arrière.

Pour la partie publique, on peut personnaliser la page d'accueil avec des widgets. Pour la partie privée, le wiki est géré par le plugins d'édition de documents texte (BuddyPress Docs).

Partage de fichiers

À la manière de ce qu'on trouve comme solution en ligne pour partager des fichiers, le plugin BuddyPress propose aux membres du site de disposer d'un espace de stockage. On peut organiser en dossiers les fichiers et les partager comme on veut : avec un seul ou plusieurs groupes, avec tout le monde ou les garder pour soi.

Un administrateur va pouvoir régler le type de fichier dont il autorise le stockage, les quotas alloués aux différents membres (éventuellement différents suivant leur rôle).

1.4 Plugins

Cette partie présente une liste de greffons à Wordpress permettant de créer un réseau social d'entreprise. les plus importants ont déjà été présentés, les autres apportent des fonctionnalités spécifiques, pas forcément primordiales.

Importants

- BuddyPress (<https://wordpress.org/plugins/buddypress/>);
- BuddyDrive (<https://wordpress.org/plugins/buddydrive/>) : partage de fichiers (privé, groupes, publics...);
- BuddyPress Docs (<https://wordpress.org/plugins/buddypress-docs/>) : édition de documents de manière collaborative;
- BuddyPress Docs Wiki add-on (<https://github.com/boonebgorges/buddypress-docs-wiki>) : permet de créer un wiki sur tout le site.

Sécurité

- Buddypress Captcha (<https://wordpress.org/plugins/buddypress-captcha/>) : Captcha à l'inscription;
- BuddyPress Members Only (<https://wordpress.org/plugins/buddypress-members-only/>) : restreint l'accès à la partie BuddyPress du site uniquement aux personnes authentifiées.

Facultatifs

- bbPress (<https://wordpress.org/plugins/bbpress/>) : intégrer un forum à Wordpress;
- BuddyPress Group Email Subscription (<https://wordpress.org/plugins/buddypress-group-email-subscription/>) : permet à un utilisateur de recevoir des notifications par mail sur les activités d'un groupe dont il est membre;
- Invite Anyone (<https://wordpress.org/plugins/invite-anyone/>) : permet d'envoyer des mails d'invitation à rejoindre le site et d'inviter des personnes pas dans sa liste d'amis à rejoindre un groupe;
- BuddyPress Message Attachment (<https://wordpress.org/plugins/buddypress-message-attachment/>) : envoyer des pièces-jointes à un message;
- Custom Profile Filters for BuddyPress (<https://wordpress.org/plugins/custom-profile-filters-for-buddypress/>) : permet aux utilisateurs de contrôler comment les liens vers leur profil sont gérés;

- CAC Featured Content (<https://wordpress.org/plugins/cac-featured-content/>) : widget pour définir jusqu'à cinq contenus à mettre en valeur sur le site ;
- BP Group Announcements (<http://commons.inabox.org/documentation/plugins/bp-group-announcements/>) : remplace la zone d'annonce pour un groupe par un onglet dans la barre d'outil.

Pour des tests

- BuddyPress Default Data (<https://wordpress.org/plugins/bp-default-data/>) : Crée des utilisateurs, groupes, discussions, etc. pour tester BuddyPress.

2 Gestion de projet avec un site Wordpress

2.1 Introduction

Afin d'enrichir le site collaboratif de l'équipe RFAI, il a été décidé d'intégrer des outils de gestion de projet. Ce document présente un état de l'art de la gestion de projet avec Wordpress. Il contient les tests de quelques plugins spécifiques pour Wordpress.

On retrouvera un résumé dans la [Section 2.4](#) présentant le tarif de chaque plugin.

2.2 Plugins Gratuits

Kanban for Wordpress

Ce plugin offre à un utilisateur authentifié la possibilité d'organiser ses tâches selon un la méthode Kanban. Une page de connexion est présentée pour accéder à un board si on n'est pas encore authentifié dans Wordpress. Un aperçu de l'interface est donné en [Figure 3](#).

On retrouve les mêmes fonctionnalités que la partie Kaban de Pilote, sauf l'organisation hiérarchique de tâches.

Un utilisateur n'a accès qu'à son board personnel, il est possible d'ajouter des collaborateurs ou de rendre un board public avec un add-on payant (\$24.95).

Orbis et Orbis Tasks

Orbis est un framework open-source (<https://github.com/wp-orbis/wp-orbis>) pour étendre Wordpress, il est à destination des entreprises. Il propose un plugin et plusieurs thèmes pour les aider dans la gestion de leurs projets, clients, etc.

L'entreprise peut créer des pages Companies qui vont la représenter ou représenter des entreprises clientes. On peut lier à une compagnie des employés (pages Persons). Et enfin, on peut rattacher des projets à une compagnie (pages Projects), avec le temps nécessaire à sa complétion.

Chaque ajout ou édition se fait avec le système de publication de posts de Wordpress, il est possible d'ajouter des commentaires sur toutes les pages.

Orbis Tasks est un add-on open-source (<https://github.com/wp-orbis/wp-orbis-tasks>) pour gérer des tâches liées à un projet. Il est rapidement présenté dans la [Figure 4](#). On peut créer des tâches, définir le temps nécessaire pour la finir, une échéance et les assigner à un utilisateur.

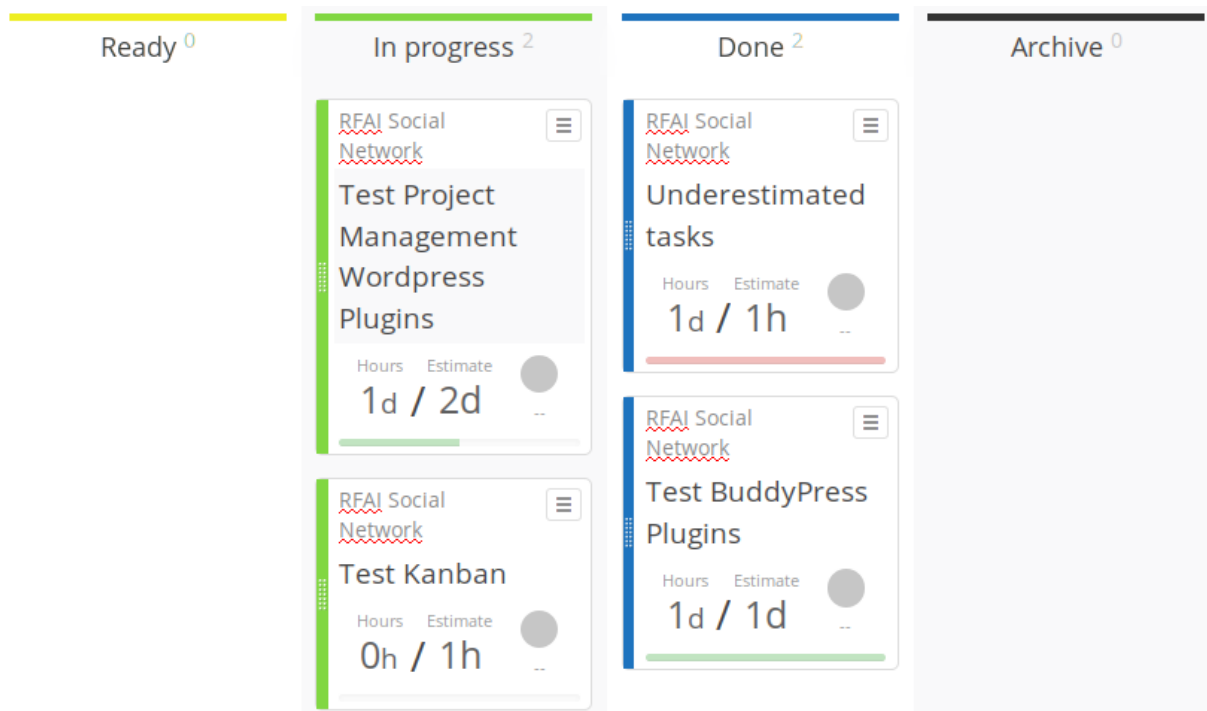


Figure 3 – Liste de tâches avec Kanban for Wordpress

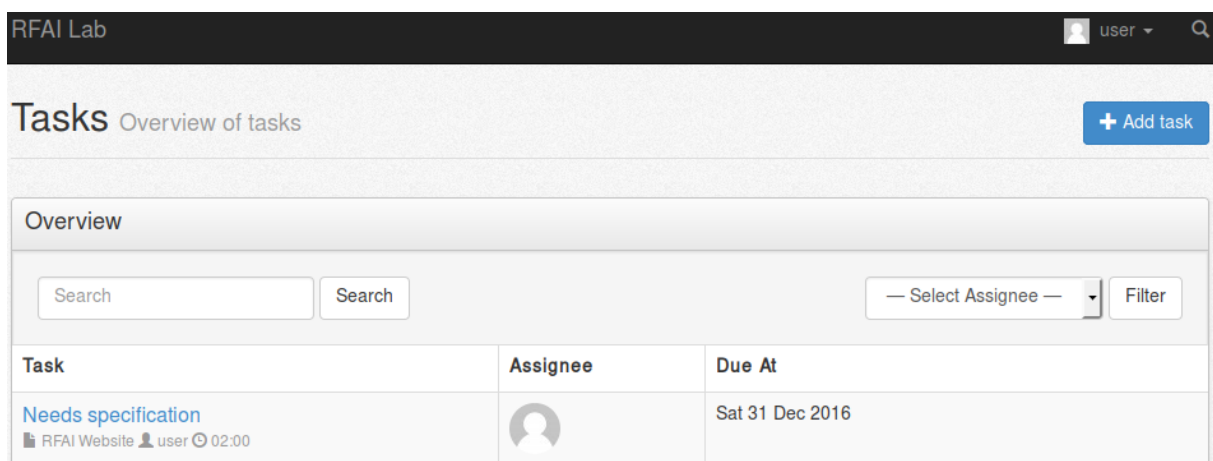


Figure 4 – Liste des tâches d'un projet avec Orbis et Orbis Tasks

La solution est assez complète pour définir des projets et des tâches.

En revanche, elle n'offre aucun moyen pour piloter et planifier un projet (par exemple, des diagrammes de Gantt). Seul un plugin, Orbis Timesheets (open source, mais vendu \$39 avec un abonnement annuel, non testé) est disponible pour tracer le temps passé sur une tâche et suivre l'évolution du temps dépensé sur un projet.

Task Manager

Task Manager est un plugin open-source (<https://github.com/Eoxia/task-manager>) développé par Eoxia une agence web de Montpellier. Il permet de gérer les tâches d'un projet et de tracer le temps consacré à leur avancement.

Côté fonctionnalités, on retrouve la création de tâches, l'affectation à un utilisateur, marquer le temps passé dessus et l'ajout de commentaire sur une tâche. On peut aussi créer des projets et affecter un temps de réalisation. Enfin, il existe une timeline qui retrace les tâches réalisées par un utilisateur.

Avec la création d'un compte pour un client, on peut lui fournir un moyen de suivre l'avancement de son projet.

CollabPress

C'est un plugin open-source qui ajoute des fonctionnalités de gestion de projet à Wordpress.

La liste des fonctionnalités est intéressante avec :

- La création d'un nombre illimité de projets, liste de tâches et tâches ;
- L'intégration avec BuddyPress ;
- L'ajout de fichiers sur des projets, tâches, ... ;
- Gestion des échéances ;
- Notifications par e-mail.

Néanmoins il n'est pas possible d'organiser ses tâches en mode Kanban et de créer un diagramme Gantt. L'autre point négatif est que le plugin ne semble plus maintenu, la dernière mise à jour date de décembre 2014.

2.3 Plugins Payants

Project Panorama

Le plugin est disponible dans une version lite, avec des fonctionnalités limitées ou dans une version premium à partir de \$59 (abonnement annuel).

Il permet de représenter de manière visuelle l'avancement d'un projet (à destination du client ou des membres de l'équipe).

Un administrateur définit les projets à afficher et rentre la date de début, de fin et l'avancement général. Il est possible de définir des étapes importantes et des phases au projet.

Un add-on payant existe pour rajouter la création de diagramme de Gantt (non testé), \$29.

Smarty Pants Client Project and Document Manager

C'est un outil pour principalement gérer des documents de manière collaborative. Des fichiers en rapport avec un projet sont téléversés et on peut les ranger dans des dossiers.

Des fonctionnalités premium telle que la définition de tâches ou de catégories à un fichier peuvent être ajoutées.

WP Project Manager

Il permet de piloter des projets en définissant des étapes importantes (milestones), des listes de choses à faire (to-do lists) et des tâches à effectuer (tasks).

Un administrateur crée un projet et définit les collaborateurs concernés. On peut aussi démarrer des discussions privées/publiques autour d'un projet. Une partie active retrace l'historique des modifications faites.

La version lite est gratuite, une version pro avec beaucoup plus de fonctionnalités (<https://wedevs.com/products/plugins/wp-project-manager-pro/#product-features>) est disponible pour \$59. Des add-ons payants existent aussi, \$29 pour s'intégrer avec les groupes BuddyPress, pareil pour gérer ses tâches en mode Kanban ou pouvoir faire des diagrammes de Gantt.

2.4 Résumé

Plugins Gratuits

Testé et fonctionnels :

- Kanban (<https://wordpress.org/plugins/kanban/>);
- Orbis & Orbis Tasks (<https://wordpress.org/plugins/orbis/>);
- Task Manager (<https://wordpress.org/plugins/task-manager/>).

Non-maintenu :

- CollabPress (<https://wordpress.org/plugins/collabpress/>).

Plugins Payants

Tous les plugins payants présentés sont vendus sous forme d'un abonnement, on paye un certain montant par an pour avoir accès aux mises à jour du plugin.

Dans le cas où l'abonnement n'est pas poursuivi, le plugin est tout à fait utilisable mais comme il ne reçoit plus de mises à jour, il peut être vulnérable ou non fonctionnel au bout d'un certain temps.

Un facteur à prendre en compte est aussi le sérieux et la solidité de l'entreprise derrière le plugin.

- Project Panorama (<https://www.projectpanorama.com/>) (à partir de \$59, 50% de réduction pour un renouvellement d'un an de la licence);
- Smarty Pants Client Project & Document Manager (<http://smartypantsplugins.com/sp-client-document-manager/>) (à partir de \$75 pour un an de licence);
- WP Project Manager (<https://wedevs.com/products/plugins/wp-project-manager-pro/>) (à partir de \$59 pour un an de licence).

3 Collaboration en entreprise avec ONLYOFFICE

3.1 Présentation générale

ONLYOFFICE est un portail web que l'on peut héberger sur son propre serveur pour mettre à disposition d'une équipe, d'une entreprise de nombreux outils lui permettant de travailler en collaboration. La solution est éditée par la société Ascensio System basée en Lettonie.

Le portail regroupe plusieurs modules qui correspondent aux principales fonctionnalités offertes :

- Documents (stockage et édition en ligne);
- Projets (gestion de projets avec définition et attribution de tâches, sauvegarde du temps passé et diagrammes de Gantt);
- CRM (pour gérer les relations clients);
- Communauté (blogs, nouvelles et sondages à l'échelle de l'équipe/entreprise);
- Personnes (gestion des membres du portail).

Il offre aussi la possibilité d'utiliser quelques outils :

- Mail (serveur de mail, agrégation de comptes, intégration avec les autres modules);
- Chat (discussion instantanée entre membres, basé sur XMPP);
- Calendrier (calendrier individuel ou de groupes, synchronisation iCal et intégration avec les autres modules).

Chaque module (à l'exception de Documents) peut être activé et désactivé en un clic, afin de n'offrir que ce dont une équipe a besoin pour travailler.

ONLYOFFICE serveur est disponible en trois versions : une communautaire, libre et gratuite, sous licence publique générale GNU Affero 3 [WWW5]. Et en version payante sous deux tarifs \$1500 et \$5000 avec un an de support et de mises à jour, un panneau de contrôle, la possibilité de retirer toute référence à ONLYOFFICE (marque blanche) et de déployer plusieurs instances du portail.

Il existe aussi une version dans le Cloud, hébergée sur des serveurs d'Amazon.

Pour finir, la société qui édite ONLYOFFICE offre la possibilité aux organismes à buts non lucratifs et académiques d'utiliser ONLYOFFICE dans le Cloud ou d'autre produits gratuitement. En échange il est demandé de placer sur le site de l'organisation une bannière du produit.

3.2 Installation

L'installation d'ONLYOFFICE Community Edition est possible par différent moyens mis à disposition sur le site de la solution :

- Conteneurs docker;
- Machines virtuelles (VMware, MS Hyper-V et Oracle VirtualBox);
- Paquets (exécutable pour Windows, paquets deb et rpm pour Linux);
- Installation en un clic (on fournit les identifiants d'accès à notre serveur et ONLYOFFICE est installé par un script);
- Lien vers les sources.

Ce document présente un test de ONLYOFFICE Community Edition installé dans une VM avec VMware.

Pour une installation sur une vraie machine, voici les pré-requis :

- CPU : dual-core 2 GHz ou mieux;
- RAM : 2 GB ou plus (avec 2 GB de swap ou plus);
- HDD : au moins 2 GB d'espace;
- Logiciels selon l'installation :
 - Docker (4 GB de RAM) :
 - Docker version 1.10 ou plus
 - Windows : 7, 8, 8.1, Server 2008(R2), Server 2012(R2)
 - MySQL Server : version 5.1 ou plus
 - Linux (noyau en version 3.10 or plus)
 - mono : version 4.2 ou plus
 - MySQL : version 5.5 (*sauf* 5.7) ou plus
 - nginx

Une fois correctement installé, on arrive sur la page de configuration par l'administrateur du portail. Puis, sur la page d'accueil du portail qui permet de sélectionner un module pour commencer à travailler avec, voir la **Figure 5**.

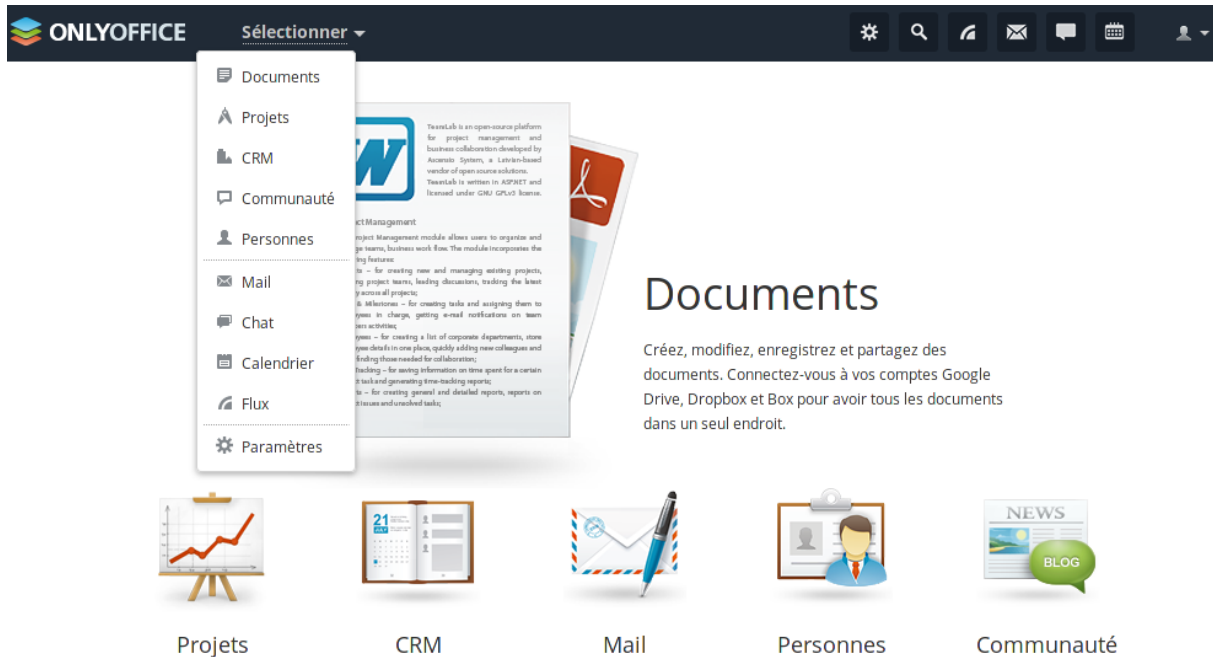


Figure 5 – Page d'accueil du portail web d'ONLYOFFICE

3.3 Sécurité

En plus d'avoir son code source libre, ONLYOFFICE possède plusieurs paramètres permettant de régler finement la sécurité du portail.

Tout d'abord, on peut configurer le portail pour chiffrer les connexions avec le protocole TLS (servir l'application en `https://`).

L'inscription au site pourra aussi être limitée aux utilisateurs avec une adresse mail d'un certain nom de domaine de confiance (comme `univ-tours.fr`). Et il est possible de définir une politique de mot de passe qui va spécifier la complexité minimale des mots de passe utilisés.

Pour aller plus loin, on pourra aussi restreindre l'accès au portail à une certaine plage d'adresse IP, typiquement un réseau local comme celui de l'école. Et aussi gérer finement les accès aux différents modules du portail pour chaque utilisateur ou groupe.

Il est à noter que lors de l'installation, l'administrateur communique une adresse mail. La société qui édite ONLYOFFICE lui envoie alors un mail via le serveur SMTP configuré par défaut (`notify.onlyoffice.info`), il peut être utilisé pour envoyer des notifications. Pour garder un contrôle sur ses données un administrateur pourra changer de serveur SMTP dans les paramètres.

3.4 Intégration

On peut choisir d'intégrer le portail avec différents outils déjà utilisés par une équipe, en voici la liste :

- Stockage dans le Cloud - Google Drive, Box, Dropbox, OneDrive, Yandex Disk, ownCloud, WebDAV ;
- Réseaux sociaux - Facebook, Google, Twitter ;
- Propre serveur SMTP pour envoyer des mails de notifications ;
- Moteur de recherche libre Sphinx pour indexer les documents.

3.5 Présentation des fonctionnalités

Documents

Ce module est le fer de lance d'ONLYOFFICE.

Il permet la gestion de fichiers, comme dans n'importe quel service de stockage en ligne. On téléverse des fichiers dans un espace privé et on peut les organiser comme on veut dans des dossiers. Lors de chaque modification, une version est sauvegardée dans un historique et il est possible de comparer les modifications et d'annuler un changement.

Chaque fichier peut être partagé à l'intérieur du portail avec les autres membres et groupes ou à l'extérieur avec un lien public. Il est possible de convertir un document (texte, classeur ou présentation) en différents formats lors de son téléchargement (par exemple, DOC vers PDF ou XLSX vers CSV).

Les documents peuvent être modifiés en ligne, il y a une très grande compatibilité avec les formats Microsoft Office (DOC, DOCX, etc.). La **Figure 6** présente l'interface d'édition de document avec un navigateur web.

ÉTAT DE FRAIS A REMPLIR A L'ISSUE DE LA MISSION									
1									
2									
3	NOM								
4	PRENOM								
5	LIEU DE LA MISSION								
6									
7	DATE DE DEPART								
8	DATE DE RETOUR								
9									
10	TRANSPORT A LA CHARGE DE L'AGENT	OUI	NON	si oui fournir les billets portant le montant					€
11									
12	NUITÉES	OUI	NON	fournir la facture de l'hôtel					€
13	rappel forfait en France 90 € en province 110 € à Paris (petit déjeuner et taxe de séjour compris)								
14									
15	REPAS	OUI	NON	fournir la facture pour chaque repas					€
16	rappel forfait en France 15,25 € par repas pris entre 11h00 et 14h00 et entre 18h00 et 21h00								
17									
18	PEAGE AUTOROUTE	OUI	NON	fournir la facture					€
19									
20	TRANSPORTS EN COMMUN	OUI	NON	fournir la facture					€
21									
22	FRAIS D'INSCRIPTION COLLOQUE	OUI	NON	fournir la facture					€
23									
24	AUTRES FRAIS			fournir les factures					€
25									
26	ATTENTION les reçus de carte ne sont pas acceptés								

Figure 6 – L'édition de document, en ligne, avec un navigateur

L'édition des documents est possible en collaboration avec différents membres, peu importe les appareils qu'ils utilisent. Un utilisateur ajoute des contributeurs au document, en précisant les droits qu'ils auront dessus, et ils reçoivent alors une notification. Chacun peut alors travailler sur

le document, deux modes d'édition existent : rapide et strict. En mode rapide, les modifications sont ajoutées en temps réel, chacun voit immédiatement le travail des autres, à la manière d'un Google Doc. En mode strict, le travail sur une partie d'un document la verrouille aux autres utilisateurs et les changements seront ajoutés lors de la sauvegarde du document.

Il est possible de discuter des changements à apporter en créant des commentaires sur le document ou en utilisant le module intégré de chat.

De manière similaire à ce que l'on retrouve dans Microsoft Office, un utilisateur peut être affecté à la revue du document et proposer des modifications. Elles sont identifiables facilement et on peut naviguer à travers toutes les propositions pour décider de leur acceptation ou non facilement.

Enfin, toutes les modifications sont sauvegardées dans un historique et il est facile de revenir à une version antérieure. Une fois une version finale du document atteinte, on peut choisir de bloquer toutes les futures modifications du document.

Les différents éditeurs (pour les textes, classeurs ou présentations) sont complets et peuvent être enrichi de plugins.

Des applications pour bureau (Windows, Linux et Mac OS) libres (AGPL version 3) et mobiles (Android et iOS, propriétaires) existent pour travailler sans connexion sur des documents.

Projets

Le module permet de piloter un projet tout au long de sa durée de vie. Lors de la création d'un projet, on définit un titre et un chef de projet. Ensuite on peut ajouter une équipe associée et des étiquettes pour mieux le classer.

On peut définir des tâches et des sous-tâches pour découper notre projet. Une tâche est attachée à un projet et on peut spécifier sa date de début et de fin ainsi que sa priorité. On peut affecter le temps utilisé à la réalisation d'une tâche à l'aide d'un minuteur ou en rentrant manuellement le temps passé.

Les jalons vont permettre de définir les dates clés de l'avancement du projet.

Une fonctionnalité importante est la création automatique de diagrammes de Gantt pour les différents projets, comme le présente la [Figure 7](#).

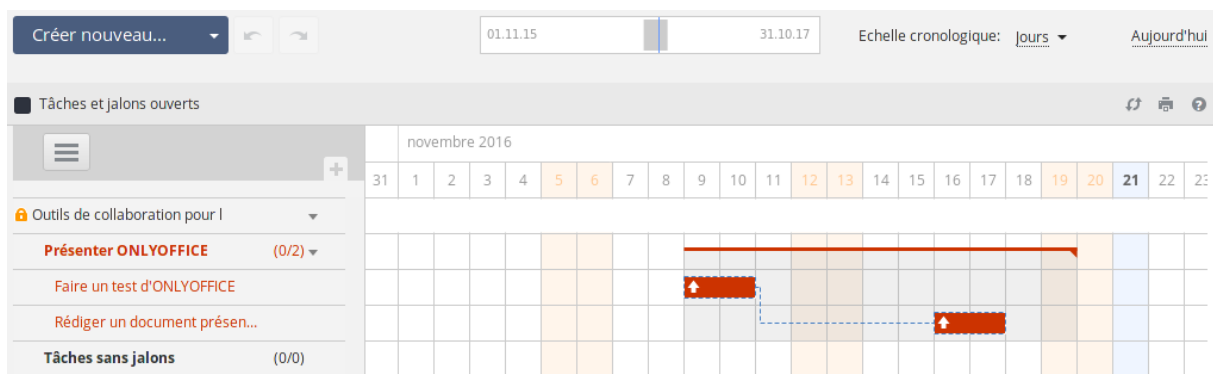


Figure 7 – Exemple de diagramme de Gantt d'un projet

Afin de se renseigner sur l'évolution des différents projets, il est possible de générer des rapports tels que les tâches ou jalons en retard, le suivi du temps ou la charge de travail.

On retrouve une intégration avec les autres modules du portail comme la visualisation des tâches sur le calendrier et la création et l'édition de document relatifs à un projet.

CRM

Ce module s'occupe de la gestion des relations avec les clients de l'entreprise.

Après la saisie de l'entreprise dans le module, les différents contacts peuvent être importés depuis un fichier CSV ou ajoutés manuellement. On peut définir pour chacun le lien avec l'entreprise. Une fonctionnalité de contact de masse est disponible pour s'adresser à plusieurs contacts dans le style d'une campagne de mail.

Le suivi des ventes est géré par l'assignation de tâches aux chargés de clientèle et en définissant des opportunités de marché.

Enfin, toutes la partie de gestion des factures peut se faire depuis le portail, de la création à l'envoi au client.

Communauté

Le module communauté propose des outils de collaboration internes à l'équipe ou l'entreprise.

Des blogs peuvent être tenus, ils vont permettre de créer des publications de manière similaire à Wordpress. Chaque publication peut être commentée.

De manière similaire, il est possible d'éditer des événements simples. Ce sont des nouvelles, ordres, annonces et sondages qui permettent de faire passer des informations et de recueillir l'avis de collègues.

Un annuaire de liens pour les utilisateurs va servir de banque de données utiles à l'équipe. Conjointement, un module complet de wiki est disponible pour créer des pages à destination des utilisateurs. Chaque page peut avoir plusieurs auteurs, les différentes versions sont sauvegardées et il est possible de joindre des fichiers à une page. Une rubrique d'aide permet d'apprendre tout ce qu'il est possible de faire avec le wiki.

Citons aussi la faculté de créer des forums avec une liste de sujets et des fils de discussion.

Une dernière partie s'occupe de rappeler les anniversaires à venir des membres (libre à eux de publier leur date de naissance) afin de respecter, ou pas, les traditions en place dans l'entreprise pour ces événements.

Personnes

Ce module permet à un administrateur de gérer les membres du portail ONLYOFFICE.

Leur ajout est soit manuel, soit par import d'une liste existante. Les contacts déjà existant sur Google et Yahoo sont pris en charge. Sinon on peut téléverser un fichier CSV au format 'e-mail,prénom,nom' ou au format CSV d'Outlook Express. La version Enterprise peut s'intégrer à un annuaire LDAP en place.

L'administrateur définit le type de l'utilisateur (administrateur, utilisateur ou invité), les groupes auquel il appartient et ses droits. Chaque utilisateur recevra alors un courriel de demande de confirmation de l'adresse mail associée à son compte.

C'est aussi ici que se passe la création et gestion des groupes d'utilisateurs du portail.

3.6 Présentation des outils

Mail

Cet outil sert comme agrégateur et permet prendre en charge plusieurs comptes e-mail d'un utilisateur, afin de regrouper ses correspondances en un seul endroit.

Un système d'étiquettes permet de trier de manière classique ses mails en fonctions des règles choisies.

L'outil s'intègre avec les autres modules de la solution en proposant, par exemple, d'ajouter des documents sauvegardés sur le portail en pièce-jointe de mails ou d'ajouter des événements annoncés dans un courriel aux calendriers en place.

Chat

La discussion instantanée entre collègues est proposée par cet outil basé sur le protocole XMPP. Il est possible d'utiliser un client web fourni par le portail ou d'utiliser son client XMPP favori (par exemple, Pidgin ou Trillian).

Une version compacte du client web est accessible depuis n'importe quelle pages du portail afin de ne rien perdre de ses discussions.

Calendrier

Le calendrier permet en un coup d'œil de suivre ses événements ou ceux de l'équipe/entreprise. Les autres modules, tels que les projets, le CRM et les anniversaires ajoutent automatiquement des événements aux calendriers concernés. Un exemple est donné avec la **Figure 8**.

Il est possible de définir un calendrier pour une équipe et de le partager avec ses membres, chacun y aura accès et pourra éventuellement le modifier. Le partage se fait en définissant pour chacun des droits d'accès au calendrier.

Finalement, un calendrier peut être publié avec flux iCal afin de le mettre à disposition d'un outil tiers. Il est alors possible de le consulter depuis n'importe quel appareil compatible et les changements seront répercutés sur le partage.

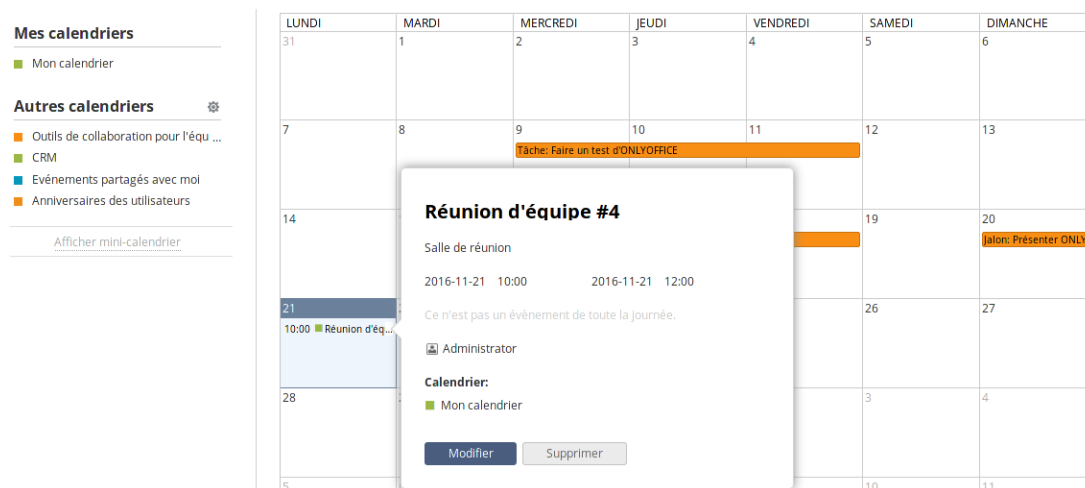


Figure 8 – Exemple de calendrier personnel

4 Listes de diffusion pour l'équipe RFAI

Les membres de l'équipe RFAI (et principalement les responsables) reçoivent un nombre important de mails concernant des communications diverses. Ces éléments sont souvent intéressants à communiquer à toute l'équipe, mais cela soulève un souci de diffusion continue de mails d'informations superflus.

Pour y remédier, nous testerons la mise en place d'une solution de liste de diffusions où chaque membre pourra déposer un mail à diffuser à l'équipe et le classer dans des catégories pour identifier son contenu. Ensuite, les membres de l'équipe RFAI s'abonneront aux listes qui les intéressent et recevront une communication choisie.

4.1 Flux RSS avec Wordpress

Wordpress met en place par défaut plusieurs flux RSS permettant au public du site de s'y abonner pour recevoir les derniers contenus publiés. Pour s'y abonner, on peut utiliser une grande variété de logiciels clients :

- Navigateurs web (sauf Google Chrome);
- Service web (Feedly, Tiny Tiny RSS, ...);
- La plupart des clients e-mails (Thunderbird, Apple Mail, ...);
- Clients RSS dédiés de bureau ou mobiles.

Les posts publiés avec Wordpress peuvent être rangés en catégorie, on peut aussi y associer une étiquette. On retrouvera chaque post dans le flux RSS du site, par exemple, pour le wiki RFAI : <http://www.rfai.li.univ-tours.fr/feed/>.

Cela devient intéressant à partir du moment où publie des posts catégorisés ou avec une étiquette. Wordpress nous offre la possibilité de retrouver un flux avec tous les posts d'une certaine catégorie, ou tous ceux avec telle étiquette.

Par exemple, on peut définir une étiquette **Publications** pour tous les posts concernant les publications scientifiques venant de l'équipe RFAI.

Lors de la création d'un post, on peut lui coller notre étiquette **Publications** avant de le publier. Ainsi notre post sera publié, donc disponible à tous, mais surtout on peut le retrouver dans le flux RSS des posts marqués **Publications**. La Figure 9 montre le flux de la catégorie **Events** du wiki RFAI.

Pour aller plus loin, on peut regrouper les posts de plusieurs catégories ou étiquettes en les séparant par une virgule.

- <http://www.example.com/category/cat1,cat2/feed>

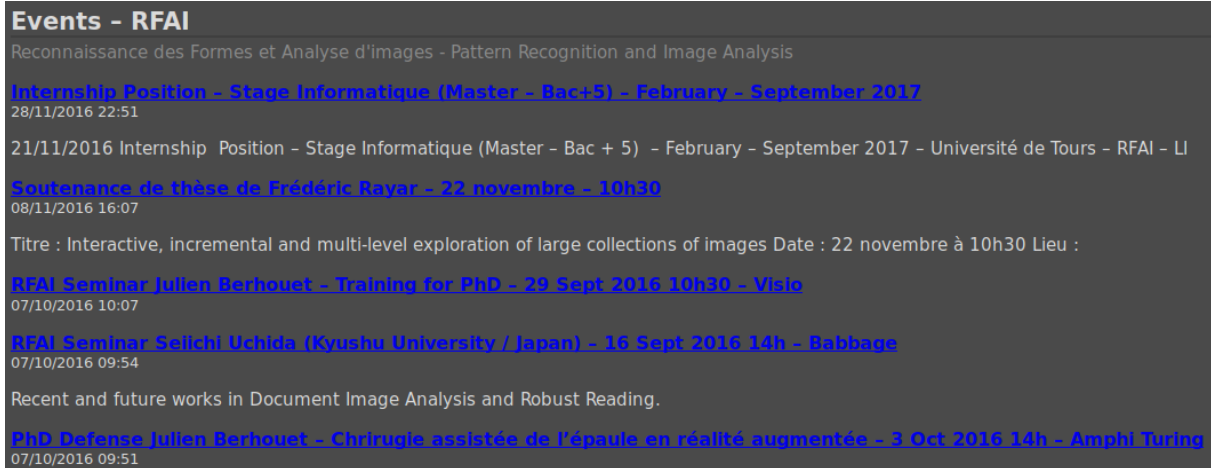
Ou en les ajoutant à la fin de l'URL :

- <http://www.example.com/category/cat1/category/cat2/feed>
- <http://www.example.com/tag/tag1/tag/tag2/feed>

Par exemple, pour avoir les posts des catégories **Events** et **News and positions** : <http://www.rfai.li.univ-tours.fr/category/events,news/feed/>

L'avantage d'utiliser ce système est la possibilité pour chaque utilisateur de s'abonner aux contenus qu'il souhaite. Les flux RSS vont lui permettre de récupérer les toutes derniers éléments mis à jours depuis presque n'importe quel appareil.

Le système mis en place ne nécessite pas de rajouter quoique ce soit au Wordpress pour fonctionner.

Figure 9 – Flux RSS des posts étiquetés *Events*

Couplé avec un plugin pour poster du contenu sur le Wordpress en envoyant un mail¹, on peut permettre à un membre de l'équipe de transmettre un mail qu'il a reçu aux personnes intéressées. Le titre du post sera l'objet du mail et il contiendra le contenu de la missive. En ajoutant quelques champs au mail, il sera possible de placer automatiquement des étiquettes et de classer le post.

4.2 Listes avec ONLYOFFICE

On peut tout à fait créer des listes de diffusion avec ONLYOFFICE, pour cela on va utiliser le module chat. Il permet de discuter instantanément entre collègues mais aussi de définir des salons de discussion.

Pour commencer on sélectionne l'icône de discussion avec plusieurs personnes et on rentre un nom pour notre liste de diffusion. Une illustration est disponible dans la Figure 10.

Ensuite, on sélectionne les contacts qui seront ajoutés à la liste et on peut alors envoyer un mail à tous les correspondants. Éventuellement, on peut ajouter une pièce-jointe au message.

On retrouvera l'aide proposée par ONLYOFFICE pour créer ses listes ici : <https://helpcenter.onlyoffice.com/guides/maillinglist.aspx>.

4.3 Listes de l'université

L'université possède déjà un service de liste de diffusion, basé sur Sympa (interface web : <https://listes.univ-tours.fr/>) cela semble le système le plus facile à utiliser et le plus pratique.

Le logiciel est présenté plus en détail dans la Section 4.4.

On pourra retrouver une introduction aux listes de diffusion (<https://listes.univ-tours.fr/wws/help/introduction>) ainsi qu'un guide utilisateur (<https://listes.univ-tours.fr/wws/help/user>) sur la documentation publiée par l'université.

L'outil est déjà intégré avec tous les services de mails de l'université. En quelques clics, on peut créer plusieurs listes pour les principaux sujets des communications reçues par l'équipe RFAI.

1. Par exemple, avec le plugin Post to your blog using email : https://codex.wordpress.org/Post_to_your_blog_using_email

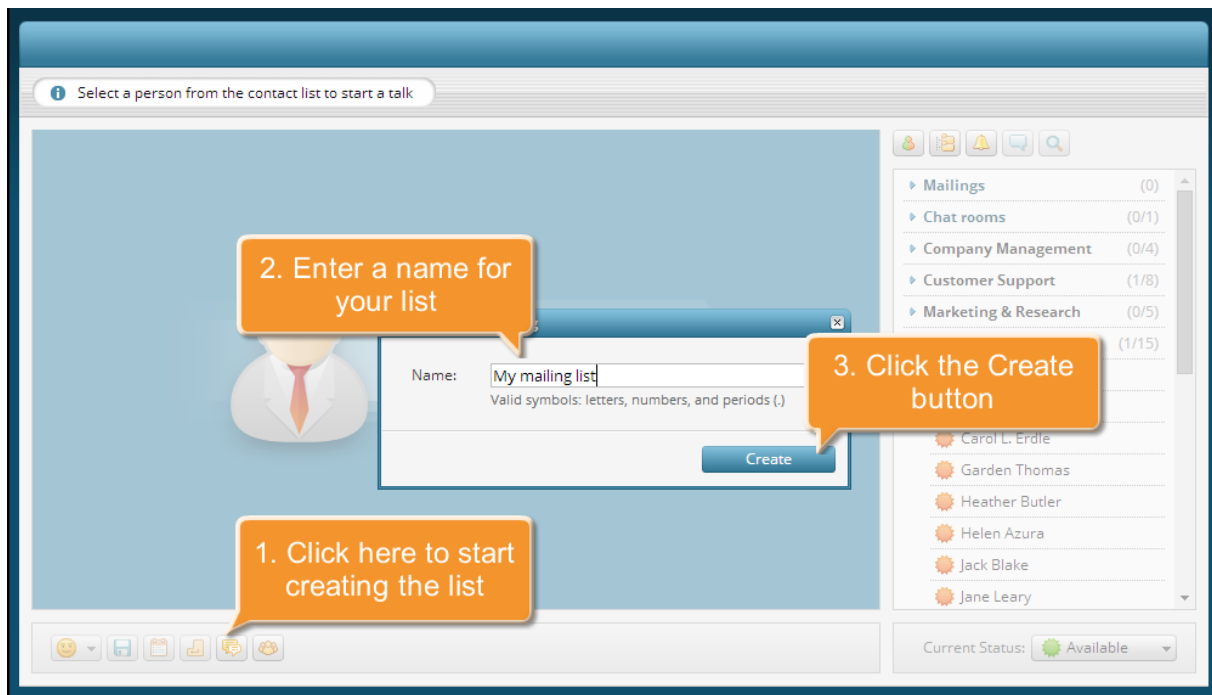


Figure 10 – Création d'une liste de diffusion (image du centre d'aide d'ONLYOFFICE)

L'authentification se fait avec les identifiants habituels pour les services de l'université (ENT, mails, etc.).

Il existe apparemment déjà une liste de diffusion pour l'équipe RFAI (rfai.li.labo@listes.univ-tours.fr), il suffit de lui envoyer un mail pour qu'il soit retransmis aux différents abonnés (voir la FAQ de la DTIC : <https://faq.univ-tours.fr/index.php?FAQ=2&THEME=15>).

4.4 Sympa

C'est un logiciel de gestion de liste de diffusion, le développement est contrôlé par RENATER, il est diffusé sous licence GPLv2 [WWW7]. La version 6.2.6 (du 15 septembre 2015) est déployée par l'université.

Une liste est créée par une personne qui deviendra son propriétaire, le nom de la liste ne devra pas contenir d'espaces, d'accents, ni de caractères spéciaux. Il a la charge d'administrer la liste en définissant les abonnés, et diverses fonctionnalités. Il peut aussi choisir de placer la liste comme visible de l'extérieur et gérer les droits d'accès.

Pour s'abonner manuellement à une liste, on peut la chercher dans l'index. Il contient toutes les listes actuellement définies. Un champ de recherche est aussi disponible. La Figure 11 présente un exemple de page d'index des listes de diffusions, ici il s'agit du début de celles de l'EPU.

Une fois abonné à une liste, on peut régler les paramètres de souscription :

- nom ;
- mode de réception (format des mails) ;
- visibilité dans la liste des utilisateurs ;
- suspendre son abonnement ;
- définir un avatar.

La transmission d'un mail aux abonnés d'une liste se fait en envoyant un mail à l'adresse de la liste, depuis n'importe quel client. Le mail est copié à tous les destinataires, il n'y a pas besoin

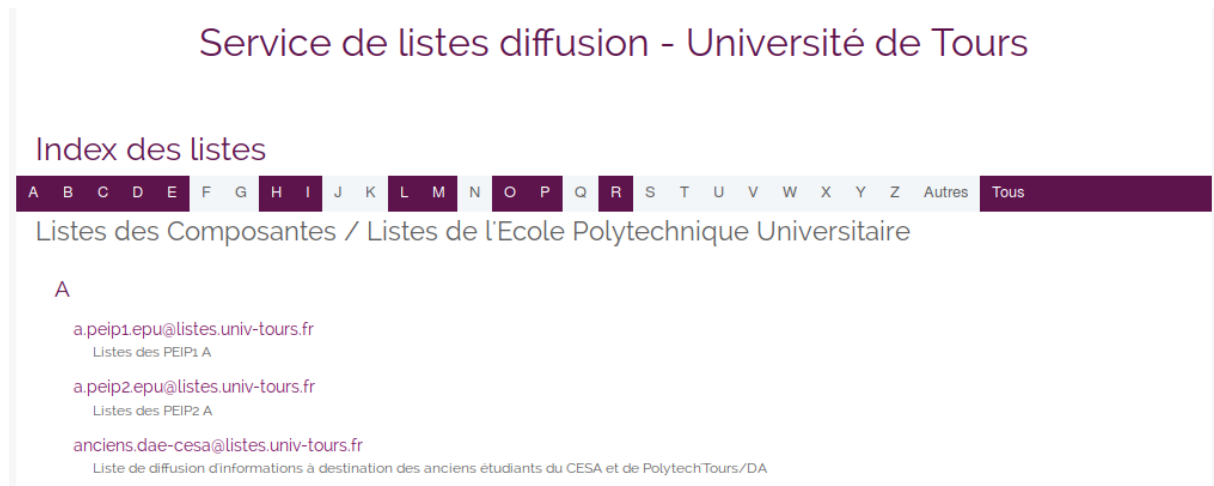


Figure 11 – *Index des listes de diffusion de l'université*

de le formater spécialement.

Sympa donne la possibilité de poster un mail depuis une page web. On ne peut pas renseigner plusieurs destinataires pour le mail, ni joindre des documents.

Le propriétaire d'une liste peut définir un espace de stockage partagé où les abonnés peuvent télécharger des documents ou en publier. Les documents sont des dossiers, des fichiers ou des signets (lien vers une page web). Le propriétaire peut restreindre l'accès à un certains fichiers ou dossiers.

Comme pour les documents, le propriétaire peut activer des archives sur une liste. Tout le contenu diffusé sera sauvegardé et consultable plus tard, afin de garder une trace des échanges.

4

Analyse et conception

La particularité de cette partie du projet est qu'elle comportait une phase importante d'état de l'art et de recherche. Au lancement du PRD, la MOA n'avait pas complètement définie ses besoins. Lors de plusieurs points avec M. Nicolas Ragot nous avons défini et exploré des solutions d'outils de collaboration. J'ai aussi participé à une réunion de l'équipe RFAI le jeudi 03 novembre 2016 où nous avons discuté des besoins et de réponses à apporter.

Les premières recherches se sont orientées vers une intégration de plugins Wordpress au site de l'équipe RFAI actuellement en ligne, ces solutions ont été présentées à la MOA et on les retrouve dans les [Section 1](#) (Chapitre 3) et [Section 2](#) (Chapitre 3).

Rapidement, le principal problème de ces outils est apparu : ils ne sont pas intégrables au wiki à cause de leur manque de fonctionnalités offertes ou de la charte de l'Université. Nous l'avons vu, la plupart des greffons offrent une seule capacité intéressante et ont une qualité inégale.

Pour proposer une solution attractive, nous avons besoin de mettre en place plusieurs modules et de former les utilisateurs à ses différents outils qui ne peuvent rien à voir entre eux. Il s'agit en fait d'un souci avec les sites Wordpress, pour enrichir un site on ajoute plusieurs greffons qui sont autant de briques que l'on pose sur les fondations. Mais les différentes briques ajoutent de la complexité à la solution et ne fonctionnent pas toujours ensemble. Par ailleurs, le fait que les greffons ne soient pas mis à jour souvent, voir plus du tout maintenu, pose de réels soucis de sécurité.

Plutôt que de proposer une solution bancaire, une autre outil a été étudié : ONLYOFFICE, détaillé dans un rapport envoyé à la MOA et dans la [Section 3](#) (Chapitre 3). Il offre plusieurs modules fonctionnant en semble pour répondre à une partie des besoins de l'équipe RFAI. La solution est en développement pas sa société-mère et une version community open-source est installable assez facilement. ONLYOFFICE permet de proposer une solution complète pour ses besoins de collaboration.

Pour donner une bonne idée de l'outil à M. N. Ragot, j'ai déployé une instance du logiciel sur une VM mise à disposition par M. M. Rousseau du service informatique. Cette VM n'est accessible que depuis le réseau de l'université est permet de tester ONLYOFFICE à l'échelle de l'équipe RFAI.

Le dernier point étudié vise à adresser une problématique majeure que rencontre l'équipe RFAI : le diffusion continue de mails d'informations. Des pistes ont été proposées dans la [Section 4](#) (Chapitre 3) et compilés dans un document transmis à la MOA. Cela n'a pas donné lieu au déploiement d'outils ni à des tests de solutions.

Deuxième partie

Nouvelles fonctionnalités sur Pilote

5

Contexte de la réalisation

Le travail présenté dans cette partie document est effectué dans une deuxième partie du PRD « réseau d'entreprise : étude et mise en œuvre pour l'équipe RFAI ». Elle est presque indépendante de la première, qui avait pour objectif d'étudier et de mettre en place des outils de collaborations pour l'équipe RFAI.

Le maître d'œuvre (MOE) de ce projet est Mathieu Boutolleau, étudiant en 5e année de l'Ecole d'Ingénieur Polytech Tours au sein du département informatique. M. Nicolas Ragot représente la maîtrise d'ouvrage (MOA).

1 Objectifs

Il s'agissait de reprendre Pilote, un outil de gestion collaborative de projet avec la méthode Kanban créée principalement par Rémi Patrizio¹ et d'ajouter de nouvelles fonctionnalités. Vous trouverez plus d'informations sur Pilote dans le rapport de PFE de Rémi Patrizio [1].

Le logiciel permet de piloter tous types de projets en proposant une interface avancée et dynamique. Les projets peuvent être découpés en corps de métiers (pour organiser les tâches selon les personnes qu'elles vont concerner) et en étapes (par exemple, en jalons temporels). Les différentes tâches peuvent se voir associer un grand nombre d'options comme des commentaires, des dates de début et de fin, un pourcentage d'avancement... Une génération de diagrammes de Gantt automatique et un calendrier permettent de planifier le déroulement du projet. Enfin, une messagerie permet aux collaborateurs de communiquer efficacement en temps réel.

La **Figure 1** présente les trois écrans principaux de Pilote. Dans l'ordre, le tableau Kanban, le diagramme de Gantt et le calendrier.

Lors de la création d'un projet avec Pilote, l'écran principal est le tableau qui contient les différentes cartes associées au projet. Pour avoir une vue d'ensemble du projet il faut faire une synthèse manuelle des indicateurs des différentes cartes ou bien se reporter au diagramme de Gantt ou au calendrier. Ainsi il manquait une page qui présente une vue plus abstraite du projet et qui permet aux décideurs d'avoir un point de vue plus général sur les gros projets. Cette

1. Pilote est né d'un projet collectif réalisé par Hamza Ayoub, Valentin Chareyre, Sofian Hamou-Mamar, Alain Krok, Wenlong Li, Rémi Patrizio et Yamine Zaidou. Il a ensuite été repris dans un projet de fin d'études par Rémi Patrizio.

page leur permet également d'évaluer rapidement différents indicateurs et métriques comme l'avancement général, les prochaines échéances ou les tâches en retards.

Un des principaux objectifs de cette partie du PRD sera de réaliser un tableau de bord qui centralise et présente des informations sur un projet. Il faut le voir comme une page de supervision générale.

Comme Pilote a vocation à être utilisé avec des projets en rapport avec l'informatique, on souhaite pouvoir l'intégrer avec des outils de gestion de code source. Par exemple on affichera dans un tableau de bord des informations en provenance d'un dépôt de code correspondant à un projet de Pilote. L'idée est de récupérer des informations d'un outil externe comme un dépôt SVN afin de présenter les derniers changements apportés dans le code. On peut aussi penser à coupler Pilote avec une solution comme SonarQube pour afficher les principales métriques du code du projet.

Il s'agit du deuxième objectif principal, il va concerner la création d'un tableau de bord permettant de centraliser des indicateurs provenant d'outils externes. La démarche est la même que pour le tableau de bord d'un projet, on veut présenter une vue d'ensemble des indicateurs du code du projet.

D'autres tâches plus annexes sont à prévoir, comme la résolution de problèmes mineurs avec le code existant de Pilote. On peut aussi envisager la traduction complète du logiciel en anglais afin d'en faire un vrai programme international. Enfin, une fonctionnalité imaginée pour la première version de Pilote était un système d'extensions. On ajoutera à Pilote un support pour que des modules créés par la communauté puissent s'intégrer au logiciel. Par manque de temps elle n'a pas vu le jour, ce PRD est l'occasion de la créer.

Afin d'assurer une installation de la nouvelle version de Pilote, un nouveau guide d'installation sera écrit. Il sera une mise à jour complète du document déjà rédigé par Rémi Patrizio à la fin de son PFE.

Comme Pilote est distribué sous licence GPLv3², le logiciel final est aussi distribué sous la même licence.

2 Hypothèses

Les deux tâches principales (le tableau de bord projet et l'intégration avec des outils de gestion de code source) sont à réaliser en priorité. Elles seront incorporées dans le logiciel final. Pour ce qui est des tâches secondaires, et notamment le système d'extensions et la traduction en anglais elles ont une priorité basse. Si la marge de temps le permet, elles pourront être développées.

3 Bases méthodologiques

Le développement des nouvelles fonctionnalités dans Pilote a été en impliquant assez fréquemment la MOA. La gestion du projet a suivi un cycle itératif afin de concentrer les efforts sur une seule fonctionnalité du logiciel à la fois. Deux itérations principales ont été prévues afin de concevoir les deux fonctionnalités principales. Des itérations plus petites ont eu lieu pour chaque fonctionnalité secondaire. En interne les différentes tâches nécessaires à la réalisation d'une fonctionnalité sont organisées dans un tableau Kanban privé³. Il permet de tracer l'avancement général et les éléments qui doivent être complétés.

2. Le texte de la licence se trouve ici [WWW8]

3. Sur un serveur que je gère, j'ai déployé l'application libre Deck pour Nextcloud : <https://github.com/nextcloud/deck>

La démarche recherchée était de présenter à la MOA une fonctionnalité complète afin de la faire valider et de passer à la suivante. Du côté du suivi du projet, des entretiens réguliers avec la MOA ont été faits afin qu'elle soit impliquée dans l'avancement général.

Afin de vérifier que les fonctionnalités développées correspondaient bien aux spécifications et répondent aux besoins de la MOA, des campagnes de tests ont été effectuées. Tout d'abord des tests unitaires ont été associés aux nouvelles fonctionnalités développées, le framework PHPUnit a été utilisé. Dans le code existant des tests sont présents, il s'agit de ceux écrits automatiquement par Symfony lors de la création du projet et ne couvrent pas du tout les différents traitements réalisés par Pilote.

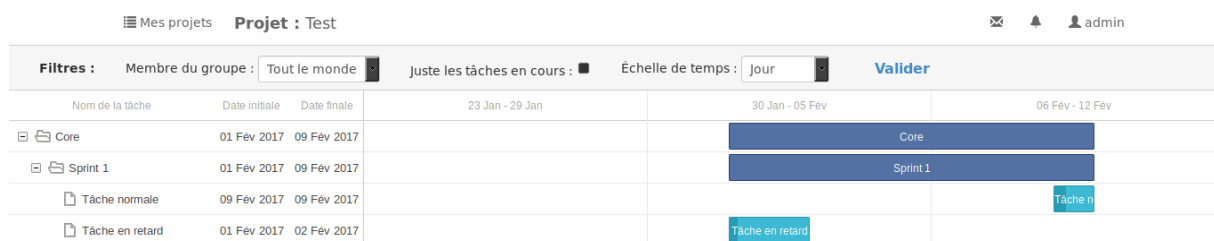
Une campagne de tests d'intégration sera aussi effectuée avant la livraison afin de s'assurer que le logiciel répond bien à la charge, on testera que les performances ne sont pas dégradées lors de la navigation simultanée par plusieurs utilisateurs. Des outils comme Selenium ou Gatling pourront être utilisés.

Des points importants de synchronisation ont été planifiés au plus toutes les deux semaines afin de discuter et d'aborder de manière complète tous les avancements du projet. Ils sont l'occasion aussi de montrer éventuellement ou en sont les développements avec une démonstration rapide, mais aussi de discuter de la suite du programme et de l'orientation des prochains jours de travail. Des points plus légers mais aussi plus fréquents peuvent être l'occasion d'aborder rapidement les tous derniers avancements, des blocages ou juste pour présenter ce qui sera effectué dans la journée. Ils sont espacés d'au plus de 2 jours et peuvent être quotidien.

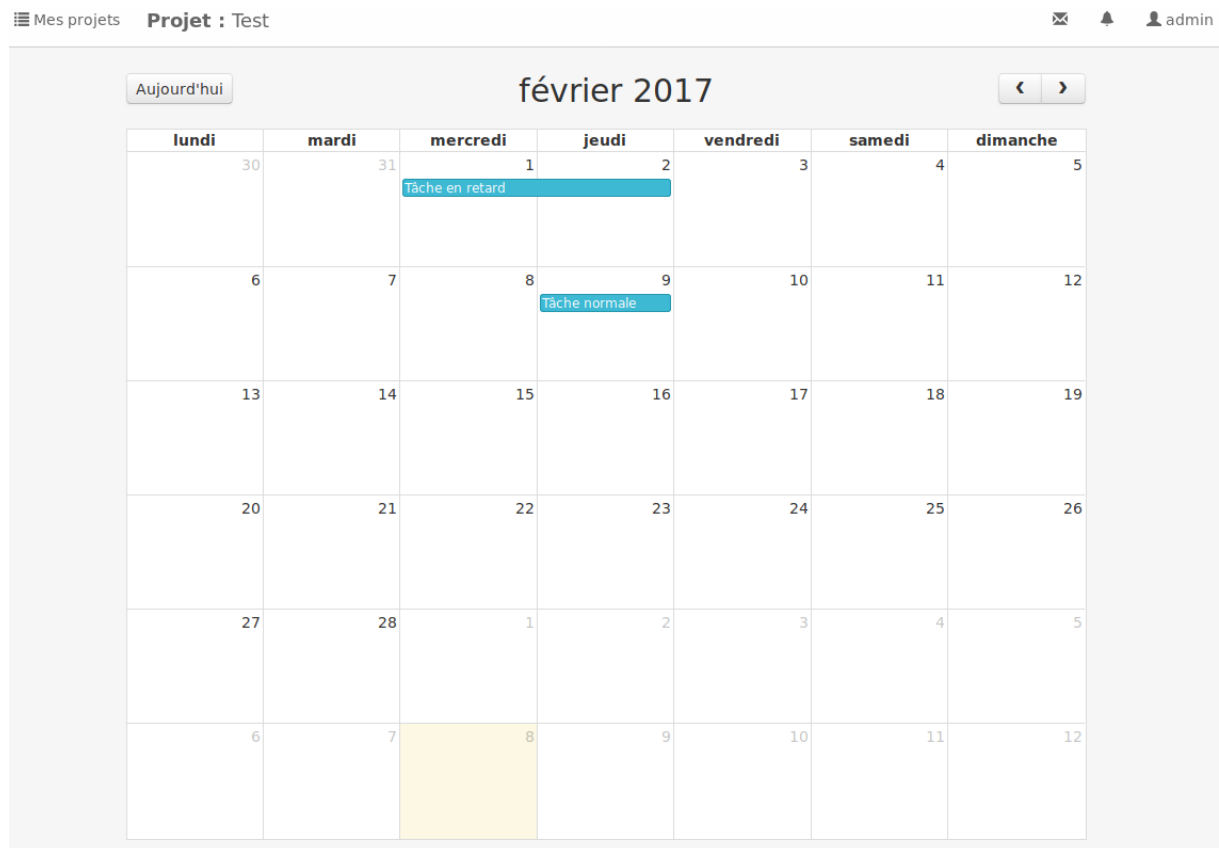
Les points importants ont eu une durée d'environ 20 à 30 minutes alors que les points rapides duraient environ 5 à 10 minutes. Enfin, un compte-rendu résumant les tâches effectuées était envoyé chaque semaine.



(a) Tableau Kanban avec les différentes tâches



(b) Diagramme de Gantt d'un projet



(c) Calendrier d'un projet

Figure 1 – Trois écrans principaux de Pilote

6

Description générale

1 Environnement du projet

Le projet est de reprendre Pilote, le code tel que livré par Rémi Patrizio à la fin de son PFE, et d'y apporter de nouvelles fonctionnalités. Il reprendra complètement l'environnement logiciel déjà existant pour l'application.

Pilote est principalement écrit en PHP en se basant sur le framework Symfony 2. Les pages présentées aux utilisateurs sont écrites en HTML en employant le moteur de template Twig (inclus dans Symfony). Elles sont mises en forme avec du CSS, principalement avec le framework Bootstrap. Il utilise aussi la bibliothèque Javascript jQuery pour l'écriture des scripts côté client. Pour finir l'application stocke ses données dans une base MySQL et elle utilise l'ORM Doctrine pour y accéder.

2 Caractéristiques des utilisateurs

Les utilisateurs de Pilote sont de trois types différents :

- Les visiteurs anonymes : ils peuvent accéder aux pages publiques du site, elles comprennent l'accueil, les pages d'authentification et d'inscription.
- Les utilisateurs authentifiés : le site est à leur disposition pour leur permettre de créer ou de supprimer des projets et d'y accéder pour les piloter. Ils peuvent aussi utiliser la messagerie instantanée et recevoir des notifications.
- Les administrateurs : ils sont chargés de veiller au bon fonctionnement du site en ayant à leur disposition une zone d'administration. Tous les projets du site leur sont également accessibles. Il choisit d'autoriser ou non les inscriptions des visiteurs sur le site.

3 Fonctionnalités du système

Les fonctionnalités principales de Pilote sont la gestion collaborative de projet avec la méthode Kanban en présentant des listes de tâches dans un tableau. Ces tâches peuvent être créées, modifiées et supprimées par un utilisateur. Un tableau de bord montrant des indicateurs sur

l'avancement du projet a été créé durant cette partie du PRD. Un diagramme de Gantt et un calendrier présente l'agencement dans le temps des différentes tâches du projet. Il peut également discuter instantanément avec d'autres utilisateurs du site au travers de la messagerie. Des notifications lui sont présentées pour lui signaler l'arrivée de nouveaux messages.

Des pages publiques permettent de présenter le site aux visiteurs et leur permettent de s'authentifier ou de s'inscrire. Une zone d'administration permet à un administrateur de superviser l'instance qu'il a déployée.

Le diagramme de la **Figure 1** est repris de celui fait Rémi Patrizio dans son rapport de PFE. Il détaille les différents cas d'utilisation possibles du site par les utilisateurs.

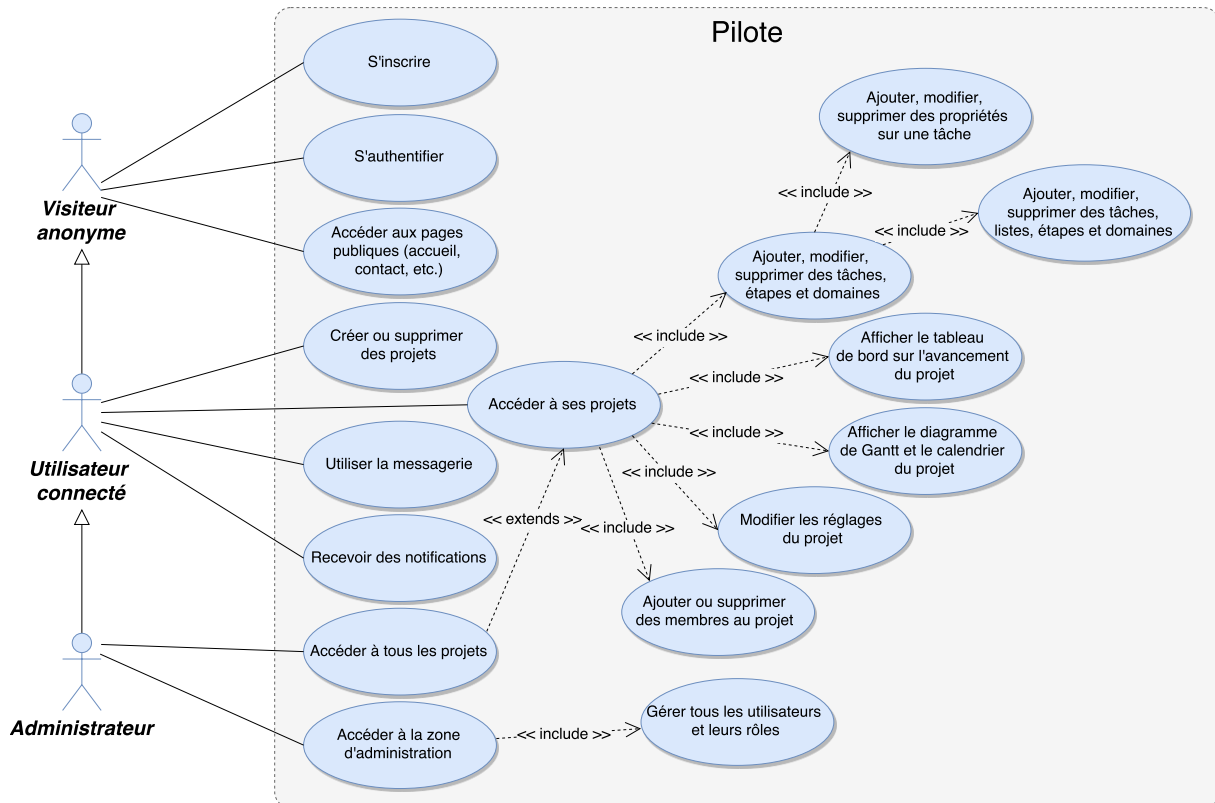


Figure 1 – Diagramme des cas d'utilisations de Pilote

4 Structure générale du système

Pilote est organisé autour de cinq éléments, la **Figure 2** schématise l'architecture générale :

- Une interface web qui permet à l'utilisateur d'interagir avec l'application et de bénéficier de ses services.
- Un serveur web qui délivre toutes les pages au client.
- Pilote, l'application en elle-même.
- Un serveur de notification, il reçoit de Pilote des notifications qu'il transmet aux utilisateurs concernés.
- Une base de données qui va sauvegarder tous les éléments en rapport avec les projets et les utilisateurs.

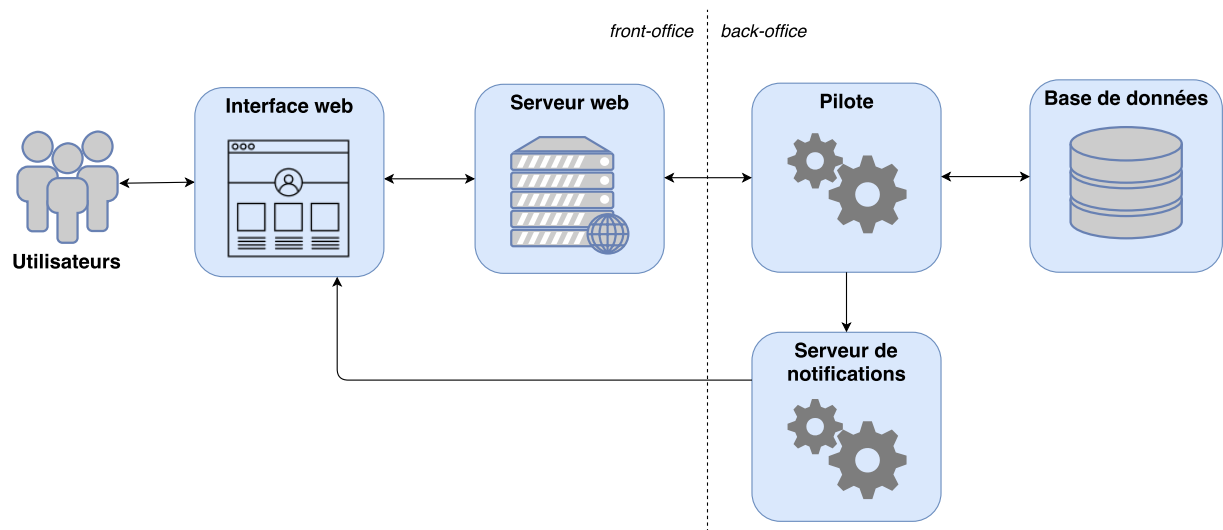


Figure 2 – Architecture générale de Pilote

7

Analyse et conception

Dans cette partie du projet nous nous sommes concentrés sur l'ajout de nouvelles fonctionnalités à un logiciel existant. L'accent a été mis sur la partie mise en œuvre et non sur une analyse ou de l'état de l'art.

La volonté derrière la création du tableau de bord de l'avancement du projet était de disposer d'une vision globale sur un projet permettant de savoir où il en est et s'il est menacé par des retards ou des échéances non-anticipées.

C'est donc naturellement que trois indicateurs se sont imposés comme indispensable à un tel outil de décision :

1. L'avancement général et détaillé du projet. Il s'agit de la moyenne de l'avancement des tâches du projet et de la date de fin définie au plus tard parmi toutes les tâches.
2. Les prochaines échéances. Il est composé des tâches dont la date de fin est prévue pour bientôt.
3. Les tâches en retard. Ce sont les tâches dont la date de fin est dépassée, un pourcentage indique le travail encore à faire.

Ces indicateurs devaient être calculées sans avoir à saisir de nouvelles informations sur les tâches du projet, tout devait être automatique. De même la palette graphique du tableau de bord doit respecter la charte en place sur le reste du logiciel. Comme il est intégré à un logiciel existant, nous allons bien entendu reprendre les mêmes technologies et faire appel aux outils de développement déjà utilisés.

Le détail des spécifications sur les nouvelles fonctionnalités de Pilote sont détaillés dans la **Partie V**.

8

Mise en œuvre

1 Architecture existante

Nous l'avons déjà abordé, la partie métier, le cœur de l'application, est écrite en PHP avec le framework Symfony. Ce framework a une approche très modulaire puisque il pousse à structurer son projet en différents modules, que l'on appelle *bundles*. Pilote est actuellement organisé autour de cinq bundles :

- **PiloteMainBundle** : Il contient toutes les pages statiques et celles générales, que l'on retrouve un peu partout (barre de navigation et pied de pages par exemple).
- **PiloteTaskerBundle** : Il contient les principales classes du modèle de l'application (*Board*, *Domain*, *Task*, *Checklist*, etc) et toutes les vues concernant l'affichage des projets (*Board*, *Tableau de bord de l'avancement du projet*, *Gantt* et *Calendrier*).
- **PiloteAdminBundle** : Il contient les pages de la zone d'administration.
- **PiloteMessageBundle** : Il contient les classes du modèle concernant la messagerie (*Message*, *Thread*, *ThreadMetadata*), et toutes les vues associées à celle-ci.
- **PiloteUserBundle** : Il hérite de *FOSUserBundle*, un bundle très utilisé et très connu, issu de la communauté Symfony. Il gère ce qui est lié aux utilisateurs (classes *User*, *Notifications*, *Pictures*) et contient quelques vues surchargées des vues fournies par *FOSUserBundle*.

La **Figure 1** montre les liens entre les différents bundles.

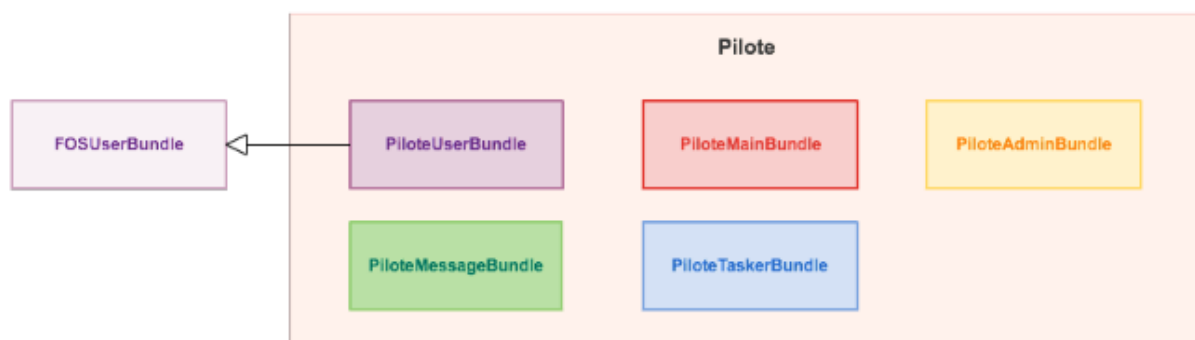


Figure 1 – Les différents modules (bundles) de Pilote, repris du rapport de PFE de Rémi Patrizio

Un projet Symfony est organisé autour du pattern MVC (modèle, vue, contrôleur) :

- Le modèle : ce sont les classes PHP correspondant à une table dans la base de données. Le lien avec la BDD est fait par l'ORM Doctrine ;
- La vue : il s'agit des fichiers HTML sans traitements PHP, les données du contrôleur sont incluses avec le moteur de template Twig ;
- Le contrôleur : il récupère des données du modèle avant de les utiliser pour générer la vue.

2 Organisation mise en place

Nous avons également un système de route qui associe une URL à un traitement particulier. En fait Symfony possède une table de routage qui fait correspondre les requêtes sur une URL à l'appel d'un contrôleur précis. Ainsi, le tableau de bord est affiché lors de la consultation de l'URL **project-dashboard/** correspondant à un projet. Par exemple, <http://localhost/pilote/web/app.php/board/1/project-dashboard/>

Le tableau de bord en rapport avec l'avancement du projet a été rajouté au bundle existant *TaskerBundle*. J'ai donc créé un nouveau contrôleur qui hérite du contrôleur, déjà existant, s'occupant des tâches d'un projet (*TaskerController*). Lui-même hérite du contrôleur par défaut du framework Symfony, sobrement appelé *Controller*.

C'est dans ce contrôleur que sont effectué le recueil des informations sur toutes les tâches d'un projet (méthode *retrieveAllTasks()*) ainsi que le calcul des différents indicateurs (méthodes *computeGeneralAndDetailedProgress()*, *computeGeneralAndDetailedEndDates()*, *computeNextDeadlines()* et *computeOverdueTasks()*).

La **Figure 2** contient le diagramme de classe du contrôleur de mon tableau de bord.

Une fois nos indicateurs réunis, nous pouvons passer à la construction de la page qui sera présentée à l'utilisateur. Pour cela nous avons une vue dédiée qui se chargera de recueillir les données en provenance du contrôleur et de les mettre en forme.

Cette vue est écrite en HTML, avec le moteur de template Twig, il s'agit d'une syntaxe reposant sur des balises qui seront interprétées pour générer la page finale. Ces balises font références aux données issues du contrôleur et peuvent définir des séquences de contrôle, comme des blocs si-alors-sinon, ou des filtres pour les traiter efficacement.

Ma vue hérite de la vue principale de Pilote, celle nommée *base* situé dans le bundle principal *PiloteMainBundle*. Elle reprend des éléments communs avec les autres vues, comme la mise en forme du titre ou les feuilles de styles générales.

La **Figure 3** présente le diagramme de classe de cette vue.

3 Résultats

Pour l'affichage des indicateurs, j'ai mis l'accent sur les indices visuels. J'ai souhaité que les données présentées soient facilement accessibles et identifiables. De nombreux codes couleurs permettent de quantifier une donnée. Par exemple, une tâche très en retard sera annoncée avec une étiquette rouge, signifiant un danger. Si un indicateur ne dispose pas des données suffisantes pour être calculés, on ne l'affiche pas (par exemple, aucune tâche en retard).

On peut voir le tableau de bord dans sa version finale avec la **Figure 4**. Nous avons nos trois grands indicateurs l'un après l'autre, dans leur conteneur respectif. Si nous voulons plus de détails sur l'avancement moyen du projet ou sur sa date de fin, on peut cliquer sur les liens présents. L'interface passe de repliée à dépliée et présente plus d'informations sur les étapes du projet, un exemple est donné avec la **Figure 5**.

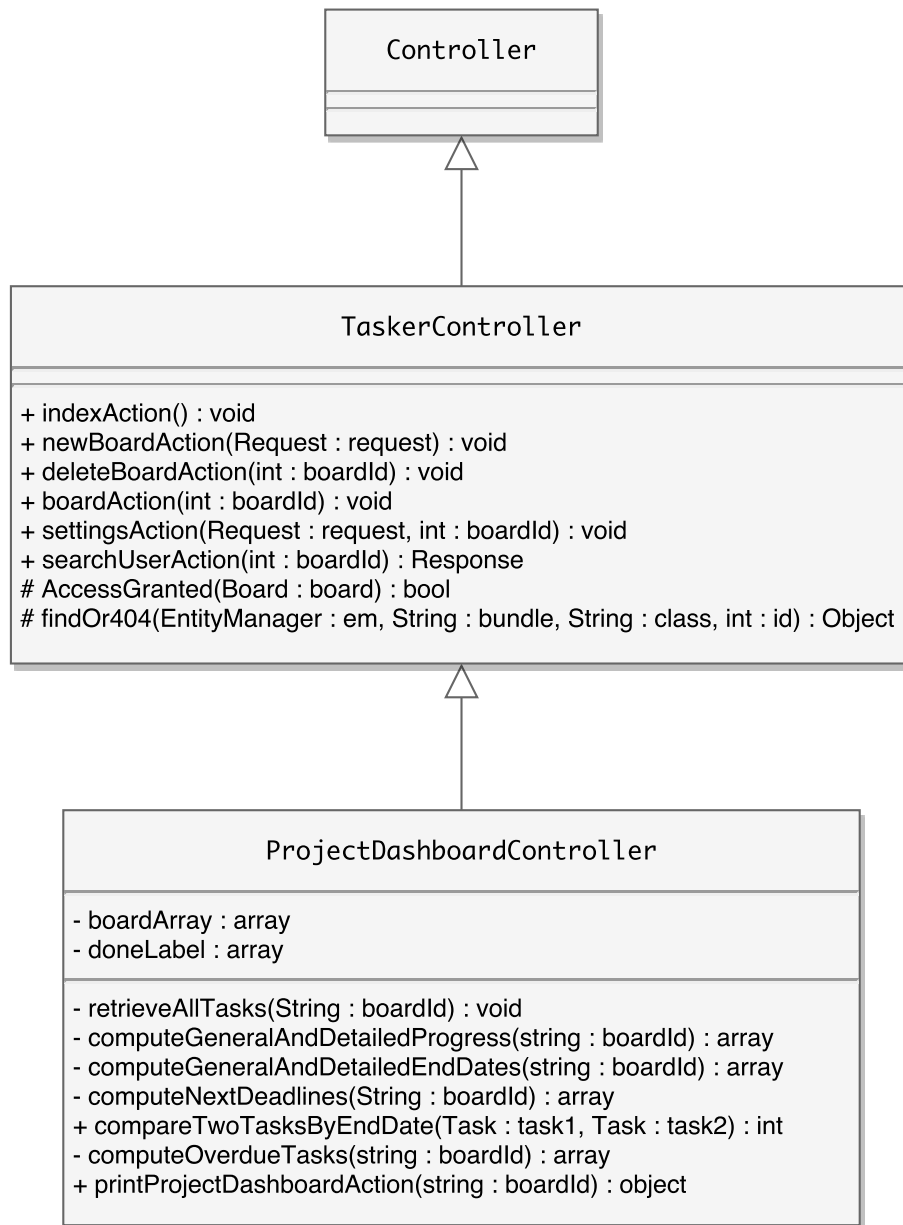


Figure 2 – Diagramme de classe du contrôleur du tableau de bord

4 Gestion des versions

Lors du développement de la nouvelle version de Pilote, j'ai utilisé l'instance Redmine de Polytech et son dépôt SVN pour la gestion des versions.

Le code original est publié sous licence libre par Rémi Patrizio sur GitHub¹. Mes modifications se retrouvent aussi sous la même licence libre et sont aussi disponibles sur GitHub². Une proposition d'ajout de mon travail au code original de Pilote sera très prochainement réalisée.

1. La version originale de Pilote est disponible ici : <https://github.com/Artemis-Haven/pilote>

2. Ma version de Pilote est disponible ici : <https://github.com/Jabberock/pilote>

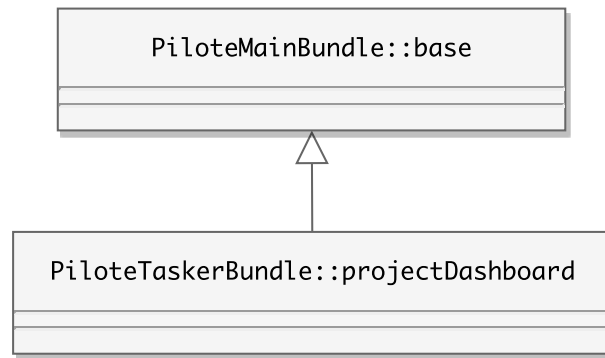


Figure 3 – Diagramme de classe de la vue du tableau de bord

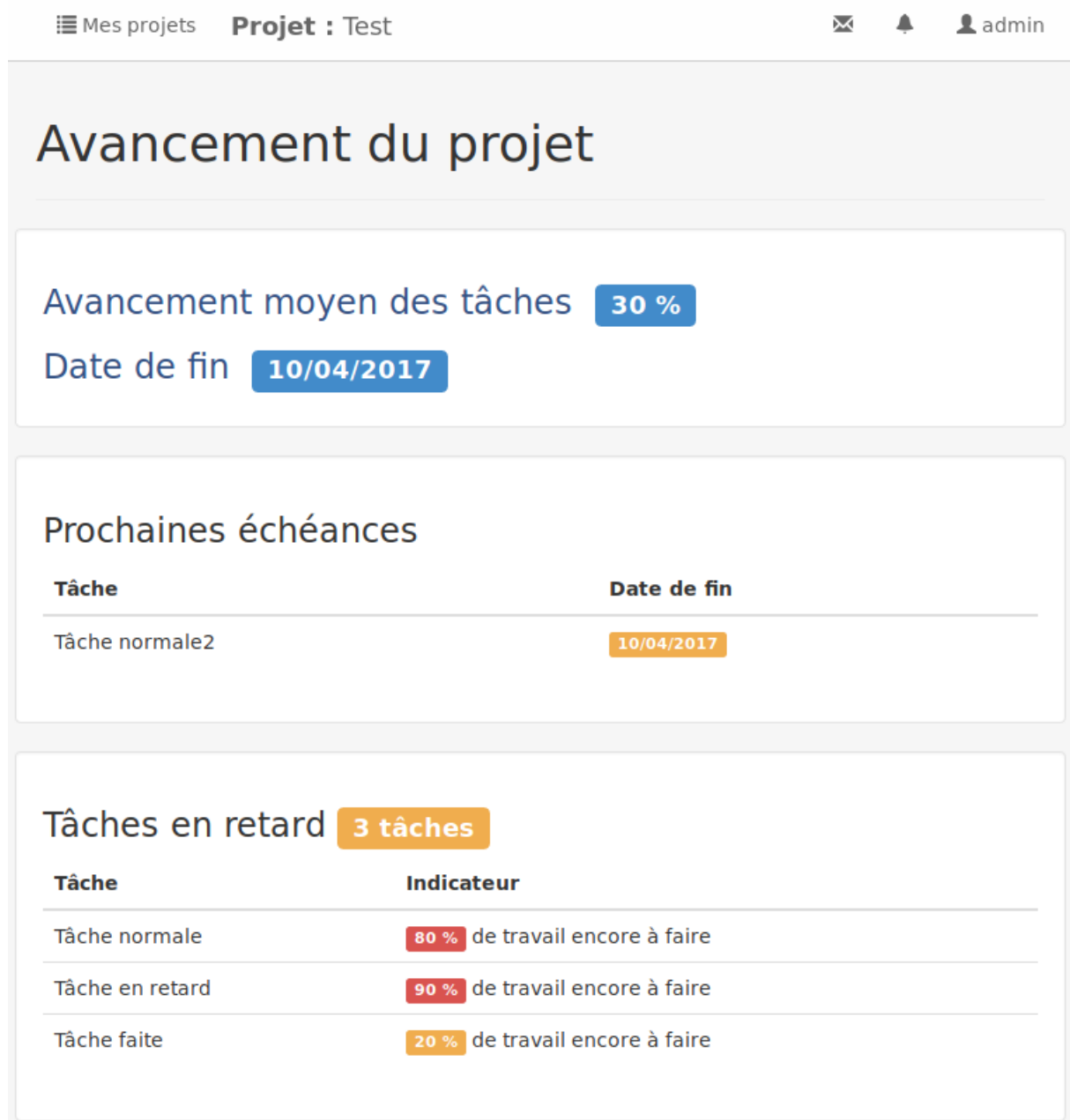


Figure 4 – Interface réalisée, en mode réplée, pour le tableau de bord

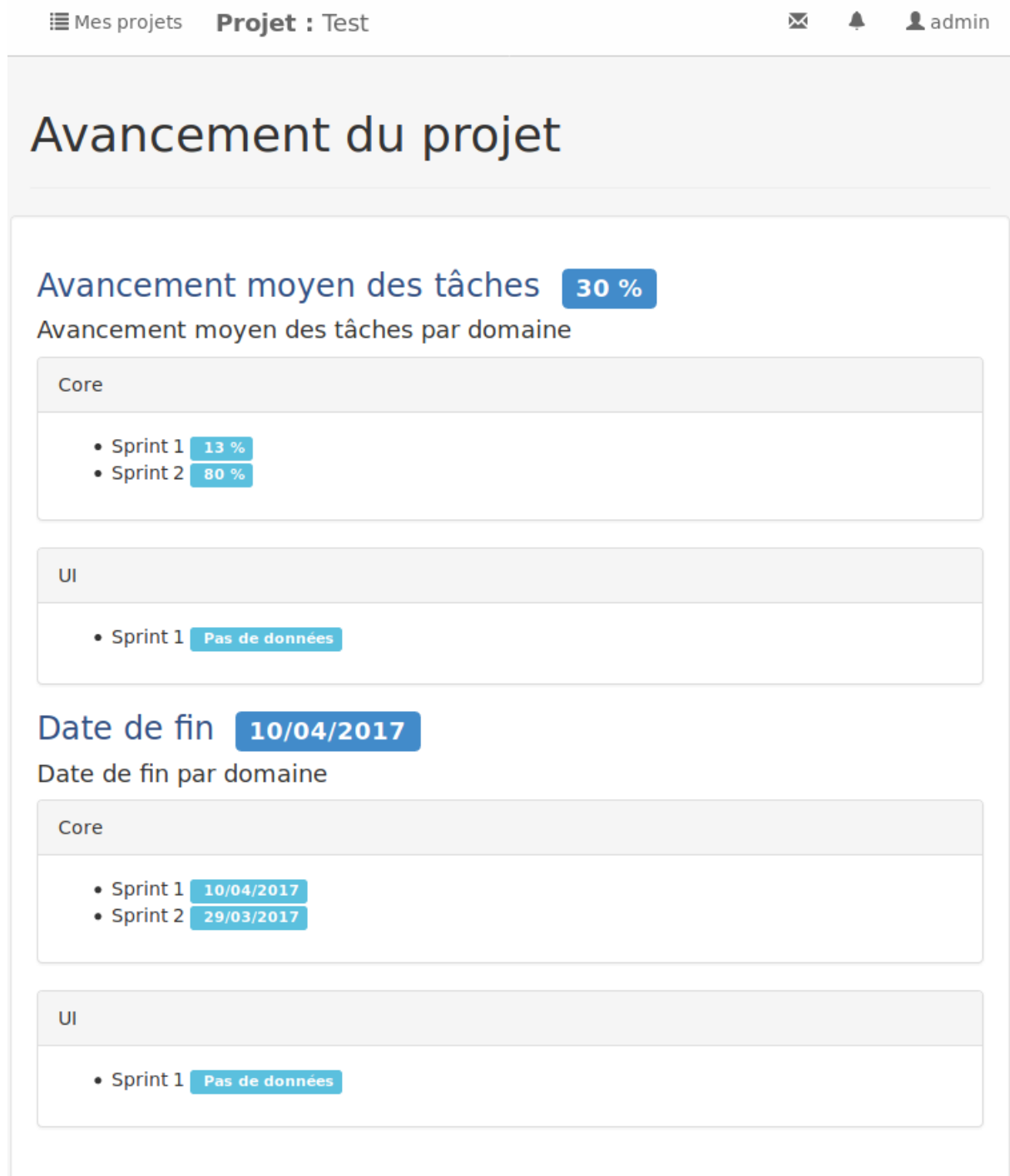


Figure 5 – Interface réalisée, en mode dépliée, pour le tableau de bord

Troisième partie

Bilan

9

Travail accompli

Nous venons de voir dans le [Chapitre 8](#) comment le tableau de bord montrant l'avancement d'un projet a été réalisé. Mais même si c'est de loin la plus importante, ce n'est pas la seule tâche effectuée au cours de cette partie du PRD.

Au lancement de la phase, j'ai pris en main le code de Pilote et je me suis formé à Symfony et son écosystème. Durant cette période, nous avons fixé avec M. N. Ragot les tâches à réaliser en fonction des besoins exprimés. Une première contribution au code a été d'apporter deux corrections. La première était que certains en-têtes rappelant la licence GPLv3 étaient mal écrits dans différents fichiers sources et étaient la cause d'exceptions levées par PHP. En effet, le format de commentaires utilisé n'était pas le bon et le texte était mal interprété. L'en-tête d'environ soixante fichiers a été modifié. La deuxième correction concernait des mappings incorrects entre la base de données et l'ORM Doctrine.

Les différentes versions des bibliothèques, frameworks et langages utilisés ont été changées pour correspondre aux dernières versions stables publiées par leurs éditeurs. Une autre priorité était de n'utiliser que des versions dont le support serait assuré pour au moins un an. Il fallait éviter ainsi de faire appel à un outil non mis à jour, avec des défauts ou des failles de sécurité. Je me suis ensuite attelé à la conception et à la rédaction des spécifications pour le tableau de bord du projet. Le plus gros du travail a été, logiquement, passé sur le codage et la validation du tableau de bord.

Une dernière phase s'est concentrée sur la préparation de la livraison de la nouvelle version du logiciel en reprenant et en complétant la documentation déjà existante. Cette documentation se trouve en annexe, dans l'[Annexe H](#).

10

Qualité

La volonté de produire un code de qualité a en partie été motivée par le fait qu'il allait être publié et mis à disposition. De même le PRD est l'occasion de mettre en application tous les aspects abordés à l'école.

Tout d'abord l'utilisation pertinente des commentaires permet d'expliquer correctement les traitements implémentés. J'ai voulu que la compréhension de mes différentes fonctions soient la plus facile possible. Dans un autre registre, les commentaires déjà présents dans Pilote étaient au format PHPDoc. Cet outil permet une génération automatique de la documentation des classes et méthodes. Il est l'équivalent pour le langage PHP de la Javadoc pour le Java. La documentation produite est présente dans le dossier **sources/apidoc/**. On peut en voir un aperçu avec la figure **Figure 1**.

Une campagne de tests a également été menée afin de s'assurer que le code produit était conforme aux spécifications et qu'il se comportait correctement. Tout d'abord des tests unitaires ont permis de vérifier que la seule méthode "métier" du nouveau contrôleur n'avait pas de défauts. Pour cela j'ai fait appel à la bibliothèque PHPUnit, elle fournit des outils pour réaliser des tests unitaires. Pour faire une comparaison, elle est similaire à ce que propose la bibliothèque JUnit pour Java.

Le reste des tests a été fait plus manuellement, dans le sens où je n'ai pas trouvé de moyen simple de le faire avec des tests unitaires. J'ai donc créé plusieurs projets qui contiennent différentes tâches afin de tester le plus de cas possibles. Par exemple, avec des tâches plus ou moins en retard, ou encore des tâches finies... Ces tests ont permis de déceler des problèmes avec les anciennes versions de certaines portions du code.

La **Figure 2** présente le tableau de bord d'un projet qui ne possède aucune tâche.

Le dernier point concernant la qualité est l'analyse du code par un outil dédié, SonarQube. Une fois installé, il peut examiner les sources d'un projet et de sortir une évaluation sur plusieurs critères de son code, comme les bugs et vulnérabilités ou encore la duplication. C'est un outil très puissant qui permet d'avoir rapidement des indications sur les points faibles du code. Par exemple, il va indiquer qu'une fonction est trop complexe et qu'elle devra être découpée en plusieurs morceaux simples. Pour le PHP, pas moins de 126 points sont examinés lors des analyses.

J'ai lancé l'analyse par SonarQube du code PHP de Pilote. Je me suis concentré sur les fonctionnalités principales et les résultats sont présentés dans la **Figure 3**.

\Pilote\TaskerBundle\Controller



ProjectDashboardController

Contrôleur principal des pages importantes : - Page listant les projets - Barres de navigation - Board d'un projet - Réglages d'un projet - etc.

..

Summary

Methods	Properties	Constants
✓ indexAction() - newBoardAction() - deleteBoardAction() - boardAction() - settingsAction() - searchUserAction() - compareTwoTasksByEndDate() - printProjectDashboardAction()	No public properties found	No constants found
⚠ AccessGranted() - findOr404()	No protected properties found	N/A
❗ retrieveAllTasks() - computeGeneralAndDetailedProg_ - computeGeneralAndDetailedEnd_ - computeNextDeadlines() - computeOverdueTasks()	\$boardArray \$doneLabel	N/A

Figure 1 – Documentation du contrôleur du tableau de bord, créée avec PHPDoc

On peut voir que les résultats sont excellents sur tous les points, comme la fiabilité, la sécurité et la maintenabilité du code, sauf pour la couverture par les tests unitaires. En effet, Pilote ne comporte que très peu de tests unitaires, j'ai écrit les seuls présents dans le projet et ils ne sont pas correctement localisés par le scanner de SonarQube.

Si on fait un focus sur le contrôleur de l'avancement du projet, comme le présente la **Figure 4**, on remarque que SonarQube identifie quelques soucis avec le code. Il s'agit du grand nombre d'imbrications de boucles pour nécessaires aux parcours des tâches du projet et à des blocs de conditions qui pourraient être combinés au prix d'une compréhension plus difficile du code.

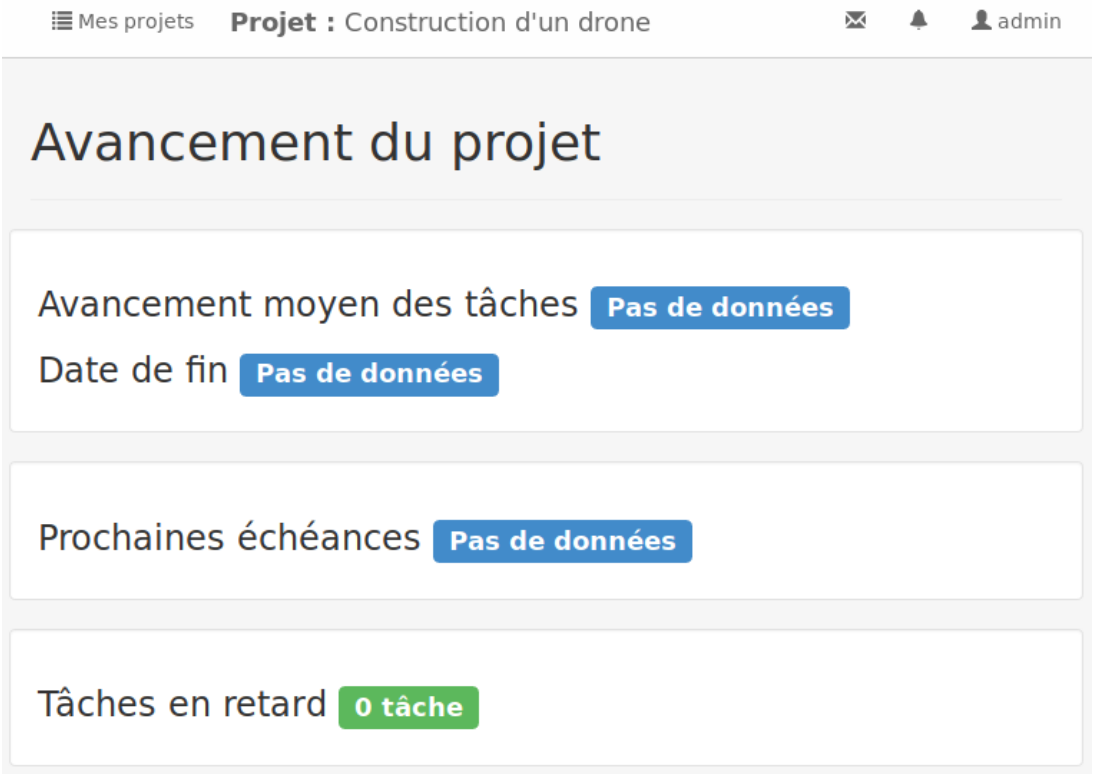


Figure 2 – Tableau de bord d'un projet vide

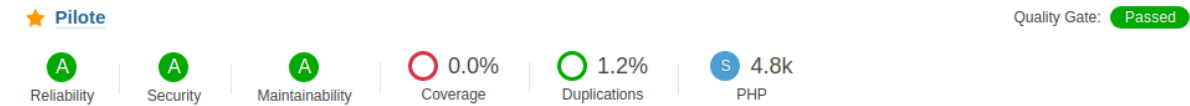


Figure 3 – Analyse par SonarQube du code PHP de Pilote



Figure 4 – Analyse par SonarQube du code du contrôleur de l'avancement du projet

11

Perspectives

Pilote est logiciel qui offre déjà de nombreuses fonctionnalités intéressantes. Il est aussi prêt pour être déployé en production et à être utilisé par une organisation de taille moyenne, même si cela n'a jamais été testé. Heureusement, il reste plein d'axes d'améliorations encore possibles et pas mal de fonctionnalités que l'on peut ajouter. De quoi remplir encore quelques PRD.

L'avenir de Pilote réside aussi dans son code source libre. Accessible à tous, il peut évoluer avec des contributions extérieures, issues de la communauté.

Idées non implémentées

Comme Pilote a vocation à être utilisé avec des projets en rapport avec l'informatique, on souhaite pouvoir l'intégrer avec des outils de gestion de code source. Par exemple on affichera dans un tableau de bord des informations en provenance d'un dépôt de code correspondant à un projet de Pilote. L'idée était de récupérer des informations d'un outil externe comme un dépôt SVN afin de présenter les derniers changements apportés dans le code. On peut aussi penser à coupler Pilote avec une solution comme SonarQube pour afficher les principales métriques du code du projet. Il s'agissait du deuxième objectif, il va concerner la création d'un tableau de bord permettant de centraliser des indicateurs provenant d'outils externes. La démarche est la même que pour le tableau de bord d'un projet, on veut présenter une vue d'ensemble des indicateurs du code du projet. Il n'a pas été réalisé à cause de la complexité trop forte et d'un manque de temps nécessaire à sa réalisation dans ce PRD.

D'autres tâches plus annexes peuvent être aussi à prévoir, comme la résolution de problèmes mineurs remontés par l'analyse de SonarQube. En se concentrant sur la partie PHP, l'outil estime à 4 jours de travail la durée de corrections de toutes les anomalies.

On peut aussi envisager la traduction complète du logiciel en anglais afin d'en faire un vrai programme international. Néanmoins cela sera un gros morceau, car beaucoup de chaînes de caractères sont cachées dans le code et il sera difficile de toutes les isoler.

Enfin, une fonctionnalité imaginée pour la première version de Pilote était un système d'extensions. On ajoutera à Pilote un support pour que des modules créés par la communauté puissent s'intégrer au logiciel. Par manque de temps elle n'a pas vu le jour, mais il s'agit là d'une tâche assez chronophage qui nécessitera une refonte d'une bonne partie du code pour rendre Pilote vraiment modulaire.



Conclusion

Nous venons de parcourir le travail d'une année scolaire dans ce Projet de Recherche et Développement autour de la mise en place d'outils de collaboration pour l'équipe RFAI.

Le PRD est une occasion de mettre en pratique des concepts abordés tout au long du cursus d'ingénieur. La première partie de ce PRD était orienté recherche avec une découverte d'outils pour la collaboration en entreprise. Ces outils ont été testés et présentés au porteur du projet, M. N. Ragot. En revanche, la deuxième partie s'est orientée vers une phase de mise en œuvre avec l'ajout de fonctionnalités à un logiciel existant, Pilote.

Une des spécificités de ce PRD est que l'équipe RFAI a su identifier ses besoins, mais n'avait aucune piste sur les solutions techniques à apporter. Ainsi, mon travail a été, dans la première partie du projet, d'être une force de proposition en présentant des solutions. Ce travail a été possible en faisant différents points avec le coresponsable de l'équipe RFAI, M. N. Ragot ainsi qu'une réunion avec tous les membres de l'équipe. Dans cette partie, l'accent a été mis sur l'analyse et la rédaction de plusieurs documents d'états de l'art.

Le travail sur Pilote m'a permis d'acquérir de nombreuses connaissances. Que ce soit sur le développement avec le framework Symfony, auquel je ne m'étais jamais vraiment approché, ou sur la rédaction de documentation claire.

Un autre point très formateur est l'évaluation continue du projet tout au long de l'année. Par exemple, la présentation de ma mise en œuvre dans le cadre de l'évaluation dédiée pousse à savoir présenter son code et ses choix de modélisation. Les autres rapports et soutenances passées permettent elles aussi de faire valider son travail pour qu'il soit le plus réussi.

Annexes

Quatrième partie

Outils de collaboration pour l'équipe RFAI

A

Description des interfaces externes du logiciel

Interfaces matériel/logiciel

La solution sera hébergée sur un serveur de l'université. Une connexion internet et un navigateur web sont nécessaires pour accéder au portail web. Il est aussi possible d'accéder à la solution via un accès VPN depuis n'importe quel autre machine connectée à Internet.

Interfaces homme/machine

L'ensemble des outils mis à disposition de l'équipe RFAI sont regroupés en trois modules : le wiki, l'outil pour la gestion des mails d'équipe et celui pour le partage et l'édition de document. L'interface principale du wiki RFAI est celle qui présente le contenu mis à disposition du public par l'équipe, nous l'avons vu c'est un site web fait avec WordPress.

Reliée à la première, une interface servira à authentifier un membre de l'équipe RFAI ou un administrateur afin de déverrouiller l'accès à d'autres parties du site.

Une autre interface, fourni par le framework WordPress, permettra alors à l'équipe RFAI de publier et de modifier du contenu pour le site. Une dernière interface permettra à l'administrateur de gérer et de maintenir le site, il s'agit du panneau de configuration fourni avec le site.

Pour le partage de document et leur édition en ligne, on propose une interface reprenant ce qui se fait dans divers outils du marché, une maquette est présentée en **Figure 1a**, dans la **Figure 1**.

Les fichiers présents sur le serveur sont affichés sous forme de liste, il est possible de les modifier, de les supprimer ou de les partager via trois boutons au-dessus de la liste. Mais aussi de les télécharger avec un bouton placé au niveau du document ou d'en sélectionner plusieurs avec des cases à cocher.

Une barre de navigation permet de parcourir les fichiers et un champ de recherche est présent pour en isoler quelques-uns. Sur la gauche, on retrouve deux boutons pour créer un document ou pour en envoyer un sur l'espace de stockage, les différents dossiers seront aussi navigables par une liste au même endroit. Enfin, un bouton donnera accès à la page de configuration du module.

Quant à l'interface d'édition des documents en ligne, elle présente le contenu du fichier en son centre, avec une large place pour son contenu. Au-dessus, on retrouve un bandeau de

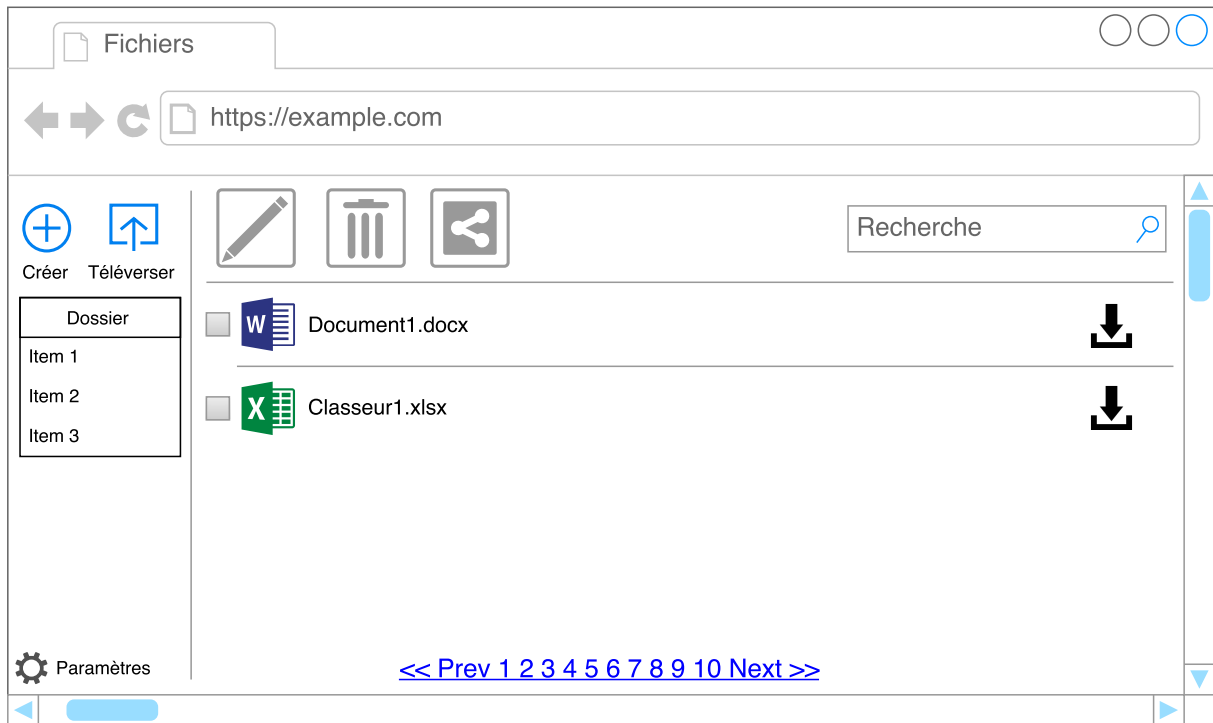
paramètres permettant de modifier des éléments du document. Une maquette est proposée avec la **Figure 1b**.

L'interface de l'outil de diffusion des e-mails n'a pas été définie.

Interfaces logiciel/logiciel

Afin de mettre en place les outils de collaboration avec le site existant, plusieurs interfaces entre logiciels existeront, la plupart étant réseau.

Les sites du LI et le wiki de l'équipe sont accessibles via le protocole HTTP (Hypertext Transfer Protocol) servi par un serveur web. On peut préciser aussi que le dépôt des pages personnelles des membres de l'équipe RFAI sur l'espace commun se fait par FTP (File Transfer Protocol). Les disques dur partagés par le réseau de l'école seront joignables avec le protocole SMB (Server Message Block), ce protocole va être utilisé pour permettre l'accès de ces espaces de stockage aux membres de l'équipe mais aussi aux différents outils mis en place.



(a) Maquette de l'interface du module de stockage de fichiers



(b) Maquette de l'interface du module d'édition de document

Figure 1 – Maquette d'interfaces

B

Spécifications fonctionnelles

Les différentes fonctionnalités à prévoir pour le site web collaboratif permettent de répondre à une demande de l'équipe RFAI pour améliorer la façon de travailler de ses membres. Cette partie présente en détail chaque fonctionnalité prévue et détaille un certain nombre d'éléments la concernant.

Partage de calendrier

Sur le site WordPress de l'équipe RFAI a été incorporé un calendrier commun. L'agenda est accessible publiquement et sert à annoncer les événements à venir concernant l'équipe. La **Figure 1** montre le calendrier disponible sur le wiki.

Il est possible de retrouver le calendrier au format iCalendar pour s'abonner et l'intégrer à tous ses appareils compatibles.

On proposera à l'équipe un espace central pour modifier le calendrier et les changements seront répercutés sur le calendrier partagé.

Diffusion de mails

Pour limiter la transmission systématique à toute l'équipe de chaque mail de communication générique reçu par les différents membres, une solution de diffusion sélective sera mise en place, c'est un outil prioritaire.

Ce système est similaire à un ensemble de liste de diffusion de courriels. Les collaborateurs de l'équipe RFAI s'abonnera à différents flux, correspondant à différents types de communication que l'équipe a l'habitude de recevoir. L'outil sera alors chargé de communiquer les mails qu'il reçoit aux membres concernés.

Lors de la réception d'un e-mail à destination des membres de l'équipe RFAI, une personne peut identifier le propos et le ranger dans des catégories pré-définies. Ensuite, la personne va le relayer sur un espace dédié. La diffusion se fera aux membres abonnées aux catégories définies du mail.

Les détails techniques de la solution n'ont pas été définis.

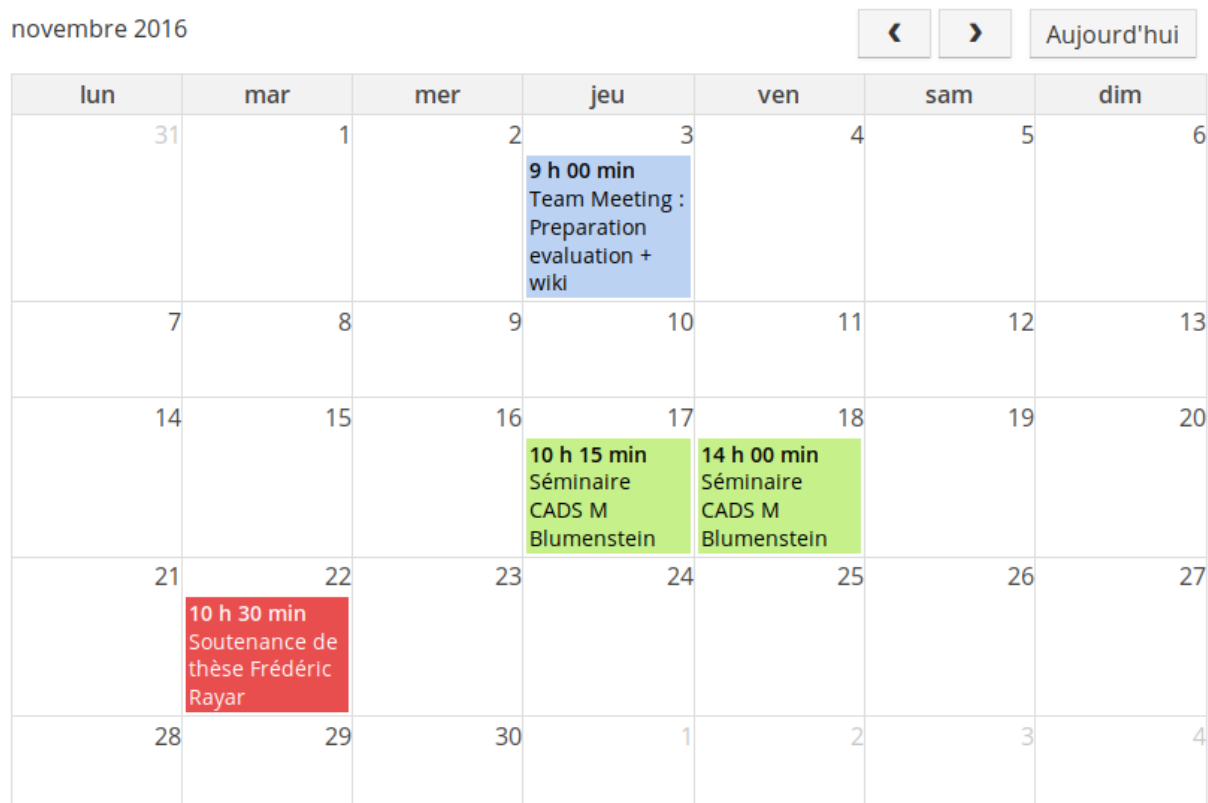


Figure 1 – Calendrier d'équipe, publiquement affiché sur le wiki RFAI

Partage de documents

Avec ces différentes missions, l'équipe RFAI est amenée à éditer et gérer des fichiers, la plupart étant des documents. Dans une volonté de partage simple entre ses membres, l'équipe emploie des espaces d'hébergements de fichiers disponibles sur internet. Cela pose un souci de confidentialité, car on ne sait pas toujours où et comment sont archivés les fichiers.

Pour reprendre la main, l'équipe RFAI souhaite disposer d'un espace de stockage interne au laboratoire qui lui permettra d'archiver et de partager ses différents fichiers, c'est une fonctionnalité prioritaire.

Concrètement, il s'agit de mettre à disposition sur une machine de l'école un espace de partage de document, et ensuite le portail ONLYOFFICE permettra de mettre à disposition ces fichiers. On retrouvera le cloisonnement avec les accès propres à chaque membre et leur espace privé de stockage.

Édition collaborative de documents

Pour rendre plus efficace le travail inter-équipe, et notamment faciliter la collaboration entre plusieurs membres, on proposera une solution pour éditer facilement un document. Éventuellement, plusieurs collaborateurs pourront éditer simultanément le même document.

Une solution intéressante est celle proposée par ONLYOFFICE qui permet depuis un navigateur, et donc depuis les principaux systèmes d'exploitation, d'éditer divers documents textes, tableurs

ou présentation. Elle s'intégrera avec le partage de document mis en place pour servir les mêmes fichiers.

Gestion de projet

L'encadrant de ce PRD, M. N. Ragot, souhaite l'intégration d'un outil lui permettant de piloter des projets. Cet outil n'a pas pour objectif initial d'être utilisé par toute l'équipe RFAI, il possède une priorité secondaire.

Avec cet outil, on peut piloter des projets de leur création à leur livraison. On retrouve la saisie d'informations générales sur un projet, comme son client, les personnes affectées à sa réalisation, ses échéances, etc.

On mettra en place un découpage du projet en tâches et leur affectation à des collaborateurs. Ces derniers auront la possibilité de venir saisir le temps passé sur une tâche lors de son avancement. Les tâches seront réparties entre jalons qui marqueront les échéances importantes de la vie du projet.

Ce module permettra de visualiser toutes les étapes d'un projet sous la forme d'un diagramme de Gantt.

C

Spécifications non fonctionnelles

Contraintes de développement et conception

Pour le développement de la solution, l'accès à un serveur permettant d'héberger les différents prototypes est nécessaire. Le service informatique de l'école met à disposition des machines virtuelles pour ce cas.

Le site existant est réalisé avec le framework WordPress, comme nous l'avons vu, une charte technique a été signée.

Contraintes de fonctionnement et d'exploitation

Performances

Il n'existe pas de contraintes particulières de performances. Néanmoins on attend du système que l'exécution de chaque tâche s'effectue en un temps raisonnable, en moins de deux secondes.

Le débit disponible pour les différentes communications réseaux est fourni et garanti par l'université.

On veille à garantir une disponibilité du site web la grande majorité du temps, en laissant la place à une suspension du fonctionnement pour maintenance ou à cause d'une panne.

Capacités

Le nombre de visite que devra pouvoir recevoir simultanément le wiki de l'équipe RFAI est le même que celui du site officiel de l'équipe héberge par l'université.

On définit un compte par membre de l'équipe RFAI, environ 39 personnes en 2016. Le site doit pouvoir gérer l'ajout de nouveaux comptes.

Le système devra être correctement dimensionné pour pouvoir gérer toutes les données de l'équipe à présenter au public et celles en rapport à la gestion des projets. Par exemple certaines bases de données disponibles ont une taille de quelques gigaoctets.

Un espace disque d'au moins 40 à 50 Go semble raisonnable. D'une part pour laisser de la place aux fichiers du système d'exploitation. Et de l'autre pour avoir de la place pour archiver les fichiers de l'équipe RFAI.

Contrôlabilité

Un suivi du fonctionnement de l'application sera assuré par le responsable technique. Il a alors accès au travers de logs et d'alertes à des informations sur les erreurs rencontrées par les applications et sur des tentatives d'utilisation malveillantes.

Sécurité

Le wiki de l'équipe RFAI étant accessible au public, une contrainte forte de sécurité existe. Il faut donc veiller à déployer des outils sûrs et dont il est possible d'effectuer des mises à jour, c'est tout l'objet de la charte technique signée avant le déploiement du site.

L'authentification d'un membre de l'équipe RFAI ou d'un administrateur doit résister à une attaque de type force brute et être correctement implémentée pour ne pas permettre à n'importe qui d'avoir accès à la partie privée du site.

Les autres solutions ont pour vocation à n'être accessible que depuis le réseau interne de l'université. Cela limite donc les contraintes de sécurité pour ces outils.

Intégrité

Lors de la migration de l'ancien wiki vers celui actuellement déployé, il a été choisi de conserver les anciens liens menant aux données. Si la solution est amenée à évoluer et qu'un changement de structure s'impose, il est nécessaire que ces mêmes liens soient maintenus.

Les données traitées par l'application sont à sauvegarder régulièrement afin de les mettre à l'abri. En cas de perte d'information, une restauration des données à un état antérieur permettra de reprendre le fonctionnement normal du site web.

Maintenance et évolution du système

Une maintenance du site web est à prévoir par le responsable technique. Cette maintenance permettra d'abord d'appliquer d'éventuels correctifs de sécurité, de permettre le bon fonctionnement de l'application et enfin de veiller à la pérennité et la disponibilité des données.

Cette maintenance est régie par la charte technique, chaque modification apportée sera validée par le responsable technique. Le travail de maintenance se doit d'être un processus rapide et les différents outils déployés doivent pouvoir être maintenu dans le temps, on cherche ici à utiliser des outils dont le support par leurs développeurs est prévu pour durer.

Cinquième partie

Nouvelles fonctionnalités sur Pilote

D

Description des interfaces externes du logiciel

Interfaces matériel/logiciel

Pilote sera idéalement hébergé sur une seule machine, reliée à un réseau local pour une utilisation interne ou à internet pour un accès global. Les différents composants nécessaires (se référer à la [Figure 2](#) (Chapitre 6)) peuvent être installés sur une même machine (pour une utilisation dans une infrastructure de taille modeste à moyenne) ou répartis sur plusieurs (pour les infrastructures les plus grandes) sans que cela ne pose de problèmes.

Interfaces homme/machine

Les vues présentées à l'utilisateur par Pilote se veulent moderne et ergonomique. Le framework Bootstrap de Twitter a été utilisé pour rendre le site élégant et adaptatif, il change automatiquement en fonction de la taille de l'écran utilisé. Voici un aperçu, avec la [Figure 1](#), du tableau listant les tâches d'un projet, il s'agit de l'écran principal de Pilote pour gérer un projet.



Figure 1 – Tableau Kanban d'un projet

Pour ce qui est des IHM liées aux nouvelles fonctionnalités, on s'efforcera de reprendre au maximum la charte graphique et les visuels déjà mis en place.

Interfaces logiciel/logiciel

Comme l'application est un site web, elle peut être consultée par les utilisateurs via le protocole HTTP, ou HTTPS pour la version sécurisée. C'est selon la configuration du serveur web associé à Pilote.

Pour le système de notification, Pilote communique avec le serveur de notifications en Node.js au travers de WebSockets, un protocole réseau. Le serveur de notifications envoie ses éléments aux clients également au travers de WebSockets.

La liaison avec la base de données se fait au travers d'un ORM, un mapping objet-relationnel, afin de représenter la base comme des objets PHP. L'ORM Doctrine est utilisé, il est intégré au framework Symfony.

E

Spécifications fonctionnelles

Tableau de bord projet

Le tableau de bord projet a pour but d'offrir des indicateurs généraux concernant un projet. Il doit permettre à un décideur de voir rapidement l'avancement général, les prochaines échéances et les tâches en retard du projet. Ces éléments sont calculés en fonction des informations de chaque tâche, ainsi ils ne nécessitent pas d'indiquer plus de renseignements pour une tâche.

Une maquette de l'interface prévue est présentée dans la [Figure 1](#).

Le tableau de bord s'articule autour de deux méthodes qui vont pour l'une calculer les indicateurs d'un projet et pour l'autre de les afficher dans une page pour l'utilisateur.

- Nom : *CalculerIndicateurs*
- Rôle : Primordial
- Spécification :
 - Entrée : Informations sur les tâches d'un projet
 - Sortie : Indicateurs sur le projet
 - Préconditions : Informations complètes et valides pour chaque indicateur (par exemple, l'avancement de toutes les tâches pour l'avancement général)
 - Post-conditions : Aucune

Cette fonction calcule les différents indicateurs d'un projet en interprétant les différentes informations renseignées sur les tâches. Par exemple leurs échéances ou progression. En cas d'élément manquant, on ne pourra pas calculer un indicateur, il ne sera pas défini.

- Nom : *AfficherIndicateurs*
- Rôle : Primordial
- Spécification :
 - Entrée : Indicateurs sur le projet
 - Sortie : Page web présentée à l'utilisateur
 - Préconditions : Indicateurs du projet calculés
 - Post-conditions : Aucune

Tous les indicateurs sont affichés à l'utilisateur dans une page web qui reprend la charte graphique de Pilote. Il s'agit juste de lire les indicateurs calculés par la méthode *CalculerIndicateurs*, en cas d'indicateur non défini, on ne l'affiche pas.



Figure 1 – Maquette de l'IHM du tableau de bord projet

F

Spécifications non fonctionnelles

Contraintes de développement et conception

Apporter de nouvelles fonctionnalités à Pilote implique de reprendre tous les éléments utilisés lors de son développement. Ainsi, j'ai utilisé les mêmes langages de programmation et les mêmes outils de développement.

Pilote est principalement écrit en PHP avec le framework Symfony 2, une phase de formation a été nécessaire afin de s'approprier à la fois le code existant et la technologie. Le jQuery est aussi utilisé pour écrire des scripts côté client, ce qui n'a pas été nécessaire dans ce PRD, et Bootstrap pour le design des pages. Le lien avec la base de données, MySQL ou MariaDB, est effectué avec l'ORM Doctrine. Pour la gestion des dépendances du projet, j'ai repris l'utilisation de Composer.

Contraintes de fonctionnement et d'exploitation

Performances

Il n'existe pas de contraintes particulières de performances. Néanmoins on attend du système que l'exécution de chaque tâche s'effectue en un temps raisonnable, en moins de deux secondes.

Pour une infrastructure de taille moyenne ou inférieure, on vise à installer tous les éléments de Pilote sur une seule machine. Rémi Patrizio précise dans son rapport de PFE que Pilote fonctionne facilement sur de petites configurations. On peut prévoir un serveur avec comme système d'exploitation GNU/Linux (macOS et Windows sont aussi possibles). Côté matériel, une configuration avec une connexion standard au réseau, un gigaoctet de mémoire vive et 20 gigaoctets d'espace disque sera suffisant.

Capacités

Pilote est très bien adapté à un nombre d'utilisateurs correspondant à une infrastructure de taille moyenne, par exemple une centaine d'utilisateurs. Les technologies utilisées sont robustes et matures ce qui permet une stabilité et des performances générales excellentes.

Contrôlabilité

L'utilisation de journaux par l'application pour détailler les incidents rencontrés permettent de contrôler le comportement de l'application.

Lors du développement, divers outils intégrés au framework Symfony 2 permettent de surveiller le fonctionnement de l'application afin de vérifier un bon déroulement des traitements ou au contraire de résoudre des problèmes.

Sécurité

Comme Pilote est une application potentiellement accessible depuis internet, la sécurité devient une préoccupation importante dans son développement. Tout d'abord, l'ensemble des dépendances utilisées par Pilote sont à jour. Elles sont également libres et très utilisées dans l'industrie, cela nous assure que leur code a été revu et que leur niveau de sécurité ne laisse pas à désirer. Un utilisateur non authentifié n'a accès qu'à des pages publiques de présentation de l'application et aux pages d'authentification et d'inscription. Les comptes administrateurs peuvent être définis à l'installation ou bien n'importe quand en définissant un nouvel administrateur ou en effectuant une promotion d'un utilisateur normal.

Le bundle FOSUserBundle est utilisé afin de stocker les différents utilisateurs dans la base de données. Il permet de bénéficier d'une grande sécurité lors de la gestion de la connexion et des enregistrements des utilisateurs. Les mots de passe sont sauvegardés en toute sécurité grâce au salage et au hachage du sésame.

Doctrine nous permet d'utiliser une couche d'abstraction autour de la base de données. Cette couche va permettre de s'assurer que toutes les requêtes faites ne sont pas malformées ou possèdent des injections.

Pour assurer la confidentialité des communications entre la machine d'un utilisateur et le serveur de hébergeant Pilote, on pourra servir les pages web avec le protocole HTTPS. Pour cela un administrateur devra configurer correctement le serveur web utilisé.

Maintenance et évolution du système

Pour assurer la possibilité d'une évolution du système par une autre personne, l'accent a été mis sur la présence d'une documentation de qualité. Elle détaille les divers aspects de Pilote, de son architecture à son installation et maintenance en passant par des commentaires dans le code pour expliquer les différents traitements réalisés. Dans l'existant, un document présente une procédure d'installation complète de Pilote, il a été repris et détaillé afin d'expliquer la démarche à suivre.

G

Gestion de Projet

Pour ce PRD, la partie gestion de Projet a été abordée de façon à rendre compte régulièrement de mon avancement à mon encadrant, M. N. Ragot. C'est pourquoi j'ai choisi de travailler avec un cycle itératif, en me concentrant sur une tâche et en la complétant avant de passer à la suivante une fois validée. Cela évite de se disperser sur plusieurs travaux en même temps et de gaspiller de l'énergie. Cela a été bien pratique sur la première partie du projet, lors des phases de recherche et d'exploration d'outils.

Ainsi, chaque fin de semaine a été l'occasion de présenter le travail effectué dans un petit rapport hebdomadaire. Il s'agit en quelques phrases de résumer le travail fourni.

Des points réguliers avec l'encadrant ont permis d'orienter le projet et de définir des tâches à traiter. Au début du PRD ces points avaient pour vocation de répondre à diverses questions et de s'assurer que la direction prise était la bonne. Dans la deuxième partie du semestre, il était plus question de montrer le résultat et de présenter les avancements effectués.

Travailler en cycle itératif permet également de capitaliser des connaissances rapidement et de maîtriser les risques autour du projet. Ainsi, lors de la deuxième partie, j'ai voulu faire le premier tableau de bord (celui sur l'avancement du projet) avant le deuxième (celui sur les outils de gestion de code), c'est pour pouvoir me concentrer pleinement sur une tâche. Comme je n'avais pas d'expérience sur le logiciel ni sur les technologies, j'ai du passé du temps à me former, temps qui au final m'a empêché de pouvoir me lancer sur le deuxième tableau de bord. En revanche j'ai pu livrer un tableau de bord projet fonctionnel et testé.

H

Documentation livrée

En plus des commentaires présents dans le code et de la description des fonctions générées avec PHPDoc, plusieurs autres manuels ont été rédigés. Ils étaient tous déjà présents dans la version de Pilote telle que livrée par Rémi Patrizio. Mon travail a été de les reprendre et de les faire évoluer.

1 Présentation de Pilote

Outils de gestion collaborative de projets

Pilote est une application web Open Source, sous licence GNU GPL v3, développée en **PHP** (*Symfony2*) et en **Javascript** (*jQuery* et *Node.js*) permettant de faciliter la gestion d'un grand nombre de projets de groupes et d'utilisateurs.

Il est le résultat de deux projets d'études à l'École d'Ingénieurs Polytech Tours.

Présentation

Création de projets

Les utilisateurs ont la possibilité de créer des projets et d'inviter d'autres utilisateurs à le rejoindre.

Affichage des tâches en listes (méthode Kanban)

Dans chaque projet, une interface simple permet de créer des tâches et des listes de tâches, et de les regrouper dans des étapes et dans des domaines plus généraux. Un simple glisser-déposer suffit à les réorganiser. Pour un projet de conception de drones par exemple, le domaine *Mécanique* contiendra les étapes *Conception*, *Assemblages* et *Tests de fiabilité* alors que le domaine *Informatique* aura plutôt les étapes *Étude de l'existant*, *Développement* et *Tests*. Ainsi les différents corps de métier qui composent un projet complexe sont regroupés au sein du même outil.

Pour chaque tâche, il est possible de définir plusieurs paramètres optionnels :

- Définir des dates de début et de fin
- Poster des commentaires
- Assigner une tâche à un membre du projet
- Définir un label (*Important, Peu important, En retard*, etc)
- Lier une pièce jointe
- Définir un pourcentage de progression
- Ajouter des listes de cases à cocher

Diagrammes de Gantt et calendrier

Afin d’afficher de manière plus visuelle d’évolution du projet, Pilote offre la possibilité d’afficher les tâches sous la forme de diagrammes de Gantt et de calendriers relativement flexibles avec des options de filtrages.

De plus, il est aussi possible pour l’utilisateur courant d’afficher un diagramme de Gantt pour toutes les tâches qui lui sont assignées sur tous les projets auxquels il est rattaché.

Messagerie

Pilote propose un système de messagerie instantanée : Une discussion de groupe est automatiquement créée pour chaque projet, mais il est possible de créer des conversations privées avec les autres utilisateurs de Pilote.

Zone d’administration

L’administrateur de Pilote a accès à une zone d’administration pour afficher et gérer l’ensemble des utilisateurs et des projets, créer un nouvel utilisateur ou vérifier différents paramètres de configuration.

Pilote étant basé sur le framework PHP Symfony2, il s’installera très facilement sur n’importe quel serveur Linux (mais peut aussi tourner sur OS X ou Windows) grâce à *Composer*.

Connexion à un annuaire LDAP

Il existe deux branches disponibles de Pilote, une classique nécessitant l’inscription des utilisateurs, et une permettant d’utiliser les comptes définis sur un serveur LDAP distant.

Il est possible de faire cohabiter les utilisateurs connectés via l’annuaire LDAP et ceux inscrits directement sur le site. L’administrateur peut aussi bloquer les inscriptions en changeant simplement un paramètre, et aussi rajouter des utilisateurs depuis la zone d’administration.

2 Manuel utilisateur

Pour commencer, sachez qu’aucune page n’est accessible aux utilisateurs non connectés. Commencez donc par vous inscrire (bouton *S’enregistrer*) ou vous connecter (bouton *Connexion*).

Création d'un projet

Sur la page d'accueil, créez un nouveau projet. Saisissez un nom et une rapide description. Une fois créé, cliquez dessus pour y accéder.

Vous pouvez voir que trois **listes de tâches** sont créées par défaut, à savoir *A faire*, *En cours* et *Fait*. Vous pouvez en ajouter en cliquant sur le bouton à droite. Cliquez sur la petite flèche à droite de leur nom pour les renommer ou les supprimer.

Ces listes sont regroupées dans un **onglet Nouvelle étape**. Vous pouvez ajouter des onglets avec le bouton + à côté, et ajouter de nouvelles listes à l'intérieur de ce nouvel onglet. Cliquez sur la petite flèche à droite de leur nom pour les renommer ou les supprimer.

Ces onglets sont eux-mêmes regroupés dans un **corps de métier**. Vous pouvez les renommer, les supprimer ou en ajouter de nouveau grâce aux boutons situés à droite du titre.

Astuce : Si vous n'utilisez qu'un seul corps de métier, nous pouvez gagner de l'espace à l'écran en cliquant sur le bouton de renommage du corps de métier et en supprimant tout le texte.

Les tâches sont **réagençables**, vous pouvez les glisser-déposer au sein d'une liste, ou bien d'une liste à un autre. De même, les listes sont déplaçables au sein d'un onglet.

Éditer une tâche

En cliquant sur une tâche, la fenêtre de détail et d'édition de la tâche s'ouvre. Cliquez sur la zone centrale pour éditer la description détaillée.

Vous pouvez ajouter plusieurs éléments à une tâche : - Des **commentaires** - Un label de **priorité** : *Haute priorité*, *Priorité basse*, *En retard*, *Fait*, etc - **Associer une personne** à cette tâche : Ainsi dans le diagramme de Gantt, il est possible de filtrer les tâches par personne. De plus, l'utilisateur retrouve ses propres tâches, regroupées par projets, sur sa page de profil. - Des **listes de cases à cocher** : vous pouvez en ajouter plusieurs pour chaque tâche - Des **dates de début et de fin** : Si seule la date de fin est renseignée, la tâche apparaîtra sous forme de jalon dans le diagramme de Gantt. Vous ne pouvez pas renseigner seulement la date de début. - Une **progression** : en pourcentage - Une **pièce jointe** : si le fichier est une image, un aperçu apparaîtra au-dessus du bouton.

Tableau de bord du projet

Le tableau de bord du projet permet d'avoir une idée précise de l'avancement général du projet. Il permet également de voir les prochaines échéances et les tâches en retard du projet.

On peut accéder au tableau de bord en cliquant sur le titre du projet, puis sur **Tableau de bord du projet**.

Les différents indicateurs sont calculés en récupérant les informations de chaque tâche, ainsi ils ne nécessitent pas d'indiquer plus de renseignements en particulier :

- L'avancement moyen des tâches est la moyenne des pourcentages de progression de toutes les tâches
- La date de fin est la dernière date parmi les dates de fin de toutes les tâches
- Les prochaines échéances sont les tâches dont la date de fin arrive bientôt

- Les tâches en retard sont celles qui ne sont pas encore terminée (qui n'ont pas une étiquette **Fait** ou qui n'ont pas une progression à 100%) et dont la date de fin est passée.

On peut faire apparaître plus de détails sur l'avancement et la date de fin de chaque domaine en cliquant sur l'indicateur associé.

S'il n'y a pas assez de données pour pouvoir calculer les différents indicateurs, le tableau de bord le précisera.

Administrer un projet

L'accès aux différentes pages du projet s'obtient en cliquant sur le titre du projet, dans la barre de menu, puis sur Configurer le projet.

Sur la page de configuration d'un projet, vous pouvez le renommer, choisir un thème de couleur, mais surtout ajouter ou supprimer des membres. Saisissez les premières lettres du pseudonyme et cliquez sur Ajouter.

Astuce : Mettre un peu de couleur dans les projets permet de mieux se retrouver sur la page d'accueil et dans le menu des projets.

Diagramme de Gantt & Calendrier

Via le menu du projet, vous avez accès au diagramme de Gantt et au calendrier. Les tâches possédant des dates de début et de fin y sont représentées.

Diagramme de Gantt

Il est possible de double-cliquer sur une tâche pour faire apparaître la fenêtre de détail et d'édition. Vous pouvez aussi déplacer les tâches pour changer leurs dates de début et de fin, ou encore définir des **contraintes de précédences** en reliant avec la souris les extrémités de deux tâches.

Trois filtres sont disponibles pour modifier l'affichage du Gantt : - Afficher les tâches de tous les membres du projet, ou seules celles assignées à un membre - Afficher seulement les tâches en cours, ainsi que les jalons à venir - Modifier l'échelle de temps

Calendrier

Ici encore, vous pouvez déplacer, redimensionner ou afficher la fenêtre de détail d'une tâche. Les jalons sont affichés en bleu foncé et les autres tâches en bleu clair.

Diagramme de Gantt personnel

Dans le menu Utilisateur, vous pouvez trouver un diagramme de Gantt regroupant les tâches qui vous sont associées dans tous les projets auxquels vous participez.

Messagerie

Le menu Messagerie contient : - Un lien vers la liste des conversations non lues - Les conversations de groupe pour chaque projet - Les conversations privées - Un bouton pour marquer toutes les conversations comme lues

Pour chaque projet, une discussion est automatiquement créée, regroupant tous ses membres.

La page d'accueil de la messagerie affiche les conversations non lues. Sur la droite, on retrouve les discussions pour chaque projet, les discussions privées et la possibilité de contacter les administrateurs.

Notifications

Vous recevrez parfois des notifications si l'on vous ajoute à un projet, assigne à une tâche, ou si l'on modifie une tâche à laquelle vous êtes associé.

Profil utilisateur

La page de profil permet de modifier son pseudonyme, son adresse e-mail et son avatar. On y trouve aussi la liste de nos projets ainsi que les tâches qui nous sont associées.

3 Manuel d'installation

Dépendances (février 2017)

Logiciels

- cURL
- Git
- PHP 5.6 ou plus, avec comme extensions :
 - php-mysql (module pour interagir avec une base de données MariaDB/MySQL)
 - php-curl (module pour utiliser cURL)
 - php-ldap (module pour accéder à un annuaire LDAP)
 - php-intl (module pour l'internationalisation)
- Apache HTTP Server, avec (**suivant la version de PHP utilisée**) :
 - libapache2-mod-php5 (**PHP 5.6**)
 - libapache2-mod-php7.0 (**PHP 7.0**)
 - libapache2-mod-php7.1 (**PHP 7.1**)
- MariaDB 10 ou plus ou MySQL 5.5 ou plus
- Node.js v6 ou plus

Frameworks

- Symfony 2.8

Testé avec

Logiciels

- PHP :
 - 5.6
 - 7.0
 - 7.1
- MariaDB :
 - 10.1
- Node.js :
 - 6.9
 - 7.4

Frameworks

- Symfony :
 - 2.8

Installation

Pour créer notre environnement de développement, nous allons installer XAMPP, Symfony et NodeJS (avec NPM).

XAMPP

Téléchargez l'installateur XAMPP (ensemble de logiciels pour la création et le lancement d'applications web) ici : <https://www.apachefriends.org/download.html>

Vous pouvez choisir la version correspondante à votre système d'exploitation (Windows, Linux, macOS).

Pour Linux, il nous reste plus qu'à changer les permissions de l'installateur...

```
chmod 755 xampp-linux-*-installer.run
```

...et à le lancer

```
sudo /opt/lampp/lampp start
```

Xdebug

Xdebug est un débogueur pour PHP, il sera utile de l'installer et de le configurer pour le développement.

Copier les informations de `phpinfo()`, en allant par exemple sur ce lien <http://localhost/dashboard/phpinfo.php>, dans le formulaire de la page <https://xdebug.org/wizard.php> pour recevoir des informations adaptées à notre machine.

Pilote

Passons à l'installation du site proprement dite.

La première étape est le choix du répertoire d'installation. Par exemple, dans un dossier du répertoire utilisateur (~/) ou /opt/htdocs/ pour le mettre dans le répertoire de XAMPP. Ici, on va installer Pilote dans le dossier ~/projects/pilote/.

```
mkdir ~/projects/pilote/ cd ~/projects/pilote/
```

On clone le dépôt des sources pour télécharger le code.

```
git clone https://github.com/Jabberock/pilote
```

Pour utiliser la branche LDAP de Pilote, il faut check-out la branche ldap au lieu de master.

Pour que notre serveur Apache installé par XAMPP puisse servir Pilote, nous allons créer un lien entre le dossier des sources et le répertoire contenant les fichiers affichés par Apache.

```
ln -s ~/projects/pilote/ /opt/lampp/htdocs/
```

On peut maintenant mettre à jour Composer, c'est le gestionnaire de paquet (fourni) qui va installer les logiciels dont à besoin Pilote.

```
cd sources/ php composer.phar self-update
```

Enfin, nous pouvons lancer l'installation de Pilote. Un script d'installation sera lancé afin de configurer le site.

```
php composer.phar install
```

Ce script vous demandera plusieurs informations afin de compléter le fichier de paramètres. À chaque fois, vous pouvez laisser la valeur par défaut en appuyant sur Entrée.

- database_driver : (pdo_mysql)
- Laissez la valeur par défaut
- database_host : (127.0.0.1)
- Idem.
- database_port : (null)
- Idem.
- database_name : (pilote)
- Le nom que vous voulez donner à la base de données. Vous pouvez laisser par défaut.
- database_user : (root)
- Le nom d'utilisateur de la base de données.
- database_password : (null)
- Le mot de passe de la base de données.
- mailer_transport : (smtp)
- Laissez la valeur par défaut
- mailer_host : (127.0.0.1)
- L'adresse IP du serveur de mail utilisé.
- mailer_user : (null)
- L'adresse mail utilisé pour envoyer des mails (l'expéditeur).
- mailer_password : (null)
- Le mot de passe associé à cette adresse mail.
- locale : (fr)
- Sert à la localisation. Actuellement, seul le bundle gérant les utilisateurs est localisé.
- secret : (ThisTokenIsNotSoSecretChangeIt)
- Ce token sert à la génération des tokens CSRF. Changez cette variable par une chaîne de caractères de votre choix.
- debug_toolbar : (true)

- Laissez par défaut
- `debug_redirects` : (false)
- Idem.
- `use_assetic_controller` : (true)
- Idem
- `router.request_context.base_url`: (null)
- Cette variable sera ajoutée au début de chaque route dans les fichiers JS.
- Si vous avez suivi ce tutoriel, il faut écrire `/app.php`.
- `notification_connexion_port`: ('8010')
- Le port de votre serveur Node.JS. Il est défini dans le fichier `sources/web/js/notifications/app.js`. Laissez par défaut.
- `disable_registration`: (false)
- Permet de désactiver les inscriptions sur le site. Cela peut être utile si vous voulez limiter les accès aux seuls utilisateurs provenant de l'annuaire LDAP.

Si vous avez choisi d'installer la branche LDAP de Pilote, vous devrez aussi renseigner les champs suivants :

- `ldap_host`: (null)
- L'adresse IP de votre serveur LDAP.
- `ldap_port`: (null)
- Le port du serveur.
- `ldap_username`: (null)
- `ldap_password`: (null)
- `ldap_baseDN`: (null)
- Exemple : `OU=Utilisateurs,OU=Services,DC=entreprise`

Base de données

Doctrine ORM est une bibliothèque qui va nous permettre d'interagir facilement avec notre base de données.

On va maintenant créer une base de données vide. Son nom est celui saisi comme nom de base de données dans l'installateur.

```
sudo php app/console doctrine:database:create
```

Le remplissage de la structure notre base de données se fait aussi avec Doctrine :

```
sudo php app/console doctrine:schema:update --force
```

Notifications (serveur Node.js)

Si Node.js est bien installé, nous avons accès à npm, son gestionnaire de paquet.

Nous allons installer socket.io pour les notifications en temps réel.

```
cd web/js/notifications npm install socket.io
```

On installe aussi pm2, c'est un gestionnaire de processus pour Node.js qui va nous aider pour lancer en production le serveur de notifications.

```
cd ../../.. sudo npm install pm2 -g
```

On peut maintenant démarrer le serveur de notifications via pm2 :

```
pm2 start web/js/notifications/app.js
```

Routes

Enfin, on génère le fichier de routes et on vide le cache :

```
php app/console fos:js-routing:dump php app/console cache:clear --env=prod chmod
-R 777 app/logs app/cache app/sessions web/uploads
```

4 Manuel de maintenance

Comment mettre à jour l'application ?

Tout d'abord assurez-vous d'être connecté en tant que *root* et placez-vous dans le dossier sources de Pilote :

```
1 sudo su
2 cd /var/www/pilote/sources/
```

Exécutez les commandes suivantes, afin de récupérer les dernières sources, relancer le serveur de notifications avec les nouvelles sources, mettre à jour la base de données et vider le cache (cela permet aux nouveautés d'être réellement visibles en ligne).

```
1 git pull
2 pm2 delete app.js
3 pm2 start web/js/notifications/app.js
4 sudo php app/console fos:js-routing:dump
5 php app/console doctrine:schema:update --force
6 php app/console cache:clear --env=prod
7 chmod -R 777 app/cache app/logs app/sessions
```

Que faire après un redémarrage ?

Tout d'abord assurez-vous d'être connecté en tant que *root* et placez-vous dans le dossier sources de Pilote :

```
1 sudo su
2 cd /var/www/pilote/sources/
```

Lancez le serveur Node à l'aide du gestionnaire de processus PM2 :

```
1 pm2 start web/js/notifications/app.js
```

5 Manuel du contribution

Contribuer ou *forker* le projet

Présentation générale

Pilote est à l'origine un projet de groupe créé par 7 étudiants en 4e année à l'École d'Ingénieurs Polytech Tours en 2014. Il a ensuite été repris l'année suivante lors du Projet de Fin d'Études de

Rémi Patrizio et en 2017 pour une partie du Projet de Recherche et Développement de Mathieu Boutolleau.

Dans la suite de ce document, le fonctionnement général de l'application web est détaillé pour quelqu'un qui souhaiterait se plonger dans le code.

Merci aux contributeurs du projet :

- Hamza Ayoub
- Mathieu Boutolleau
- Valentin Chareyre
- Sofian Hamou-Mamar
- Alain Krok
- Wenlong Li
- Rémi Patrizio
- Yamine Zaidou

Technologies utilisées

Le framework PHP **Symfony2** a été utilisé pour la majeure partie. On respecte donc le modèle MVC. Le front-end est constitué de vues en HTML utilisant **TWIG**. Les animations en Javascript et les requêtes AJAX utilisent **jQuery**.

Les notifications en temps réel utilisent un serveur **Node.js** et **Socket.io**. Le lien entre la vue, le serveur Node.js et le serveur PHP est détaillé plus bas.

Structure du projet

Pilote est constitué de 5 bundles :

- **PiloteMainBundle** : Il contient toutes les pages statiques et les templates généraux comme ceux de la barre de navigation par exemple. Il y a deux contrôleurs : un pour les pages et un pour les requêtes Ajax.
- **PiloteTaskerBundle** : Il contient les principales classes du modèle de l'application (*Board*, *Domain*, *Task*, *Checklist*, etc) et toutes les vues concernant l'affichage des projets (Board, Tableau de bord, Gantt et Calendrier). Il possède 4 contrôleurs : un pour les requêtes AJAX gérant le déplacement des tâches et des listes de tâches dans la vue du Board, un pour les autres requêtes AJAX, un pour le Gantt et le calendrier, et un dernier pour les pages.
- **PiloteAdminBundle** : Il contient les vues de la zone d'administration.
- **PiloteMessageBundle** : Il contient les classes du modèle concernant la messagerie (*Message*, *Thread*, *ThreadMetadata*), et toutes les vues associées à celle-ci.
- **PiloteUserBundle** : Il hérite de **FOSUserBundle**. Il gère ce qui est lié aux utilisateurs (classes *User*, *Notifications*, *Pictures*) et contient quelques vues surchargées des vues fournies par **FOSUserBundle**.

Pilote utilise plusieurs bundles installés automatiquement avec Composer :

- **FOSUserBundle** pour la gestion des utilisateurs.
- **FOSJSRoutingBundle** pour utiliser le système de routing de Symfony dans les fichiers Javascript.
- **NCElephantIOBundle** pour le lien entre le serveur PHP et le serveur Node.js. Il propose des fonctions simples pour envoyer des notifications depuis les contrôleurs Symfony au serveur Node.js.
- Enfin, **FR3LDAPBundle** pour utiliser un annuaire LDAP et permettre aux employés d'une entreprise de se connecter avec leurs identifiants.

Envoi des notifications

Lorsqu'une action est effectuée dans une vue, une requête AJAX est envoyée au serveur PHP. Celui-ci effectue un traitement (dans un fichier `Controller.php`) et envoie une notification au serveur Node.js grâce au Bundle `ElephantIOBundle`. Le serveur PHP est donc un client du serveur Node.js.

Voici un exemple de notification envoyée depuis un controller. Ici, on envoie la liste des utilisateurs potentiellement concernés par la notification :

```
1 $client = $this->get('elephantio_client.default');
2 $client->send('simple-notification', [
3     'html' => $monTexte,
4     'users' => $usersIds
5 ] );
```

Le serveur Node.js est constitué uniquement du fichier `web/js/notifications/app.js`. Il reçoit une notification de connexion à chaque fois que quelqu'un charge une page. Il sait donc qui est connecté et sur quelle page il est. Lorsqu'il reçoit la notification du serveur PHP, il croise sa liste d'utilisateurs connectés et celle des utilisateurs concernés par la notification. Il va ensuite envoyer une nouvelle notification à ces clients.

Voici le code permettant d'effectuer ces actions :

```
1 socket.on('simple-notification', function (data) {
2     console.log('simple-notification');
3     for (var i = 0; i < data.users.length; i++) {
4         var clients = connectedClients.getSocketsForId(data.users[i]);
5         for (var j = 0; j < clients.length; j++) {
6             clients[j].emit('notification', data.html);
7         };
8     };
9 });
```

Le fichier `web/js/notifications/notifs.js` est inclus dans chaque page web et effectue le traitement côté client. Il suffit de récupérer la notification envoyée par le serveur Node.js et de répercuter ces informations dans la vue.

Voici le code concerné :

```
1 socket.on('notification', function (data) \{
2     displayNotification(data, true);
3 \});
```

Webographie

- [WWW1] APACHE SOFTWARE FOUNDATION. *Licence Apache*. Anglais. Version 2.0. Jan. 2004. URL : <https://www.apache.org/licenses/LICENSE-2.0>.
- [WWW2] BUDDYPRESS.ORG. *Site officiel de BuddyPress*. Anglais. URL : <https://buddypress.org/> (visité le 13/10/2016).
- [WWW3] CUNY ACADEMIC COMMONS. *Site officiel de Commons In A Box (CBOX)*. Anglais. URL : <http://commonsinabox.org/> (visité le 13/10/2016).
- [WWW4] ÉQUIPE RFAI DU LI. *Wiki de l'équipe RFAI*. Anglais, français. URL : <http://www.rfai.li.univ-tours.fr/> (visité le 04/12/2016).
- [WWW5] Free Software FOUNDATION. *Licence publique générale GNU Affero*. Anglais. Version 3. Nov. 2007. URL : <https://www.gnu.org/licenses/agpl-3.0.html>.
- [WWW6] SOCIÉTÉ KOSMOS. *Site officiel de K-Sup*. Anglais, français. URL : <http://www.ksup.org/> (visité le 04/12/2016).
- [WWW7] Richard STALLMAN. *Licence publique générale GNU*. Anglais. Version 2. Juin 1991. URL : <https://www.gnu.org/licenses/gpl-2.0.html>.
- [WWW8] Richard STALLMAN. *Licence publique générale GNU Affero*. Anglais. Version 3. Juin 2007. URL : <https://www.gnu.org/licenses/gpl.html>.
- [WWW9] UNIVERSITÉ FRANÇOIS-RABELAIS DE TOURS. *Site du Laboratoire d'Informatique (LI) de l'Université de Tours*. Anglais, français. URL : <http://li.univ-tours.fr/teams/rfai-500793.kjsp> (visité le 04/12/2016).
- [WWW10] WORDPRESS FOUNDATION. *Site officiel de WordPress*. Anglais. URL : <https://wordpress.org/> (visité le 04/12/2016).



Bibliographie

- [1] Rémi PATRIZIO. « Pilote, outil collaboratif de gestion de projets ». Projet de fin d'études. Tours, France : Ecole Polytechnique de l'Université François Rabelais de Tours, 2014-2015.

Réseau d'entreprise : étude et mise en œuvre pour l'équipe RFAI

Résumé

L'équipe Reconnaissance des Formes et Analyse d'Images (RFAI) du Laboratoire d'Informatique de l'Université de Tours a souhaité mettre en place de nouveaux outils qui visaient à améliorer la diffusion d'informations et le travail collaboratif au sein de l'équipe.

Dans le cadre de la première partie de mon Projet de Recherche et Développement, j'ai effectué une étude et la mise en œuvre d'un réseau social d'entreprise pour l'équipe RFAI. Ce document présente les besoins de l'équipe, un état de l'art de ce qui est actuellement disponible sur le marché en termes d'outils de collaboration et une présentation de la solution envisagée.

Lors de la seconde partie, j'ai repris un logiciel de gestion de projet développé par un ancien étudiant de l'école, Rémi Patrizio pour y apporter de nouvelles fonctionnalités. Ce fut l'occasion de perfectionner mes connaissances dans de nombreux domaines

Mots-clés

Projet de Recherche et Développement, Équipe RFAI, Réseau social d'entreprise, Outils de collaboration, Développement logiciel, Outil de gestion de projet libre

Abstract

The Pattern Recognition and Image Analysis (in french: RFAI - Reconnaissance des Formes et Analyse d'Images) team of the Laboratory of Computer Science of the University of Tours wants to set up new tools aiming at improving the distribution of information and the collaborative work between the team's members.

For the first part of my Research and Development Project, I'm doing a study and a realization of a corporate social network for the RFAI team. This document presents the team's needs, a state of the art of what is currently available on the market for collaborative tools, and a presentation of the considered solution.

During the second part, I picked up an existing project management software, written by a former school student, Rémi Patrizio, to bring new features. It was very helpful to develop new skills in numerous fields.

Keywords

Research and Development Project, RFAI team, Corporate social network, Collaboration tools, Software development, Free, as in freedom, project management tool