

PRIM

Application web de gestion de l'UE M1-PRIM

SAVEIL NUEL

ROBIN KEMPF

DANG NGUYEN

ABDOULAYE KEITA

THOMAS MOREL (CHEF DE PROJET)

14/10/2014

Dossier d'initialisation

Ce document a pour objectif de définir le cadre du projet (contraintes, objectif, contexte) ainsi que les modalités générales de sa réalisation (méthode et outils, macro-planning).

Encadrants

- Marc PLANTEVIT
- Emmanuel COQUERY

Directeurs de projet

- Romuald THION
- Emmanuel COQUERY

TABLEAU DES REVISIONS DU DOCUMENT

03	Thomas MOREL Saveil NUEL Robin KEMPF Dang NGUYEN Abdoulaye KEITA	16/10/2014	Retrait diagrammes de séquence (cahier des charges) Rectification des contraintes de coût Modification nom de l'application
02	Thomas MOREL Saveil NUEL Robin KEMPF Dang NGUYEN Abdoulaye KEITA	15/10/2014	Correction de contenus mal positionnés Reprise plus détaillée de quelques blocs de texte Ajout de diagrammes de séquence
01	Thomas MOREL Saveil NUEL Robin KEMPF Dang NGUYEN Abdoulaye KEITA	14/10/2014	Première version du dossier d'initialisation
Révision	Auteur(s)	Date	Modifications

Table des matières

TABLEAU DES REVISIONS DU DOCUMENT.....	1
I- OBJET ET CONTEXTE	3
A- Contexte	3
B- Positionnement	3
C- Objet.....	3
II- RESULTATS ATTENDUS.....	4
A- Livrables de gestion de projet.....	4
B- Livrables techniques	4
C- Autres livrables.....	5
III- METHODE ET OUTILS	5
A- Contraintes	5
B- Méthode	5
C- Outils.....	5
IV- MACRO-PLANNING	6
A- Lots de travail.....	6
B- Phasage.....	7

I- OBJET ET CONTEXTE

A- Contexte

Le projet **Prim** doit répondre à un besoin de faciliter la gestion de l'UE M1-IF Projet de Recherche (MIF20). Il est mis en œuvre par Marc PLANTEVIT, responsable de cette UE. Aujourd'hui le déroulement de l'UE MIF20 est effectué ainsi :

- Tout d'abord, les enseignants doivent déposer leurs propositions de sujet sur une feuille TOMUSS publique présente sur le site de démonstration <http://demo710.univ-lyon1.fr/MOTUS>.
- Ensuite, les étudiants sont conviés à contacter les enseignants pour pouvoir se positionner sur un sujet ayant retenu leur attention.
- Suite au contact avec les enseignants, la candidature est validée ou refusée par ceux-ci.
- Après avoir effectué leur travail de recherche, les étudiants se verront attribuer différentes notes à partir du rapport produit, de la soutenance et de l'appréciation de l'enseignant encadrant.

B- Positionnement

Dans le fonctionnement actuel de gestion de l'UE de projet de recherche, on constate les problèmes suivants:

- L'éclatement des outils de gestion de l'UE :
 - o MOTUS, qui est utilisé pour déposer les sujets et inscrire les étudiants
 - o SPIRAL, pour le dépôt et la récupération des rapports
 - o TOMUSS, pour noter les étudiants
- L'affectation des étudiants sur MOTUS est faite par les étudiants eux-mêmes, qui ont un droit d'édition sur cette feuille TOMUSS.
- La planification des soutenances est faite à la main.
- La lecture des sujets sur MOTUS est difficile pour les étudiants car l'interface est peu ergonomique et il n'y a aucune fonction de recherche par sujet, enseignant ou mots clés.

C- Objet

Tous ces points mettent en jeu différents outils de travail. L'objectif de notre projet est donc de développer une application web pour faciliter la gestion de cette UE. Cette application devra permettre de gérer les étapes suivantes :

- L'inscription des enseignants et des étudiants
- Le dépôt de sujets
- La recherche des sujets avec différents filtres (sujet, enseignant, mots clés, ...)
- Le positionnement des étudiants sur chaque sujet
- La planification des soutenances
- L'évaluation de l'UE

II- RESULTATS ATTENDUS

A- Livrables de gestion de projet

- 26/10/14 - GdPPost-sprint 1
- 30/11/14 - GdPPost-sprint 2
- 01/02/15 - GdPPost-sprint 3

B- Livrables techniques

Pré-sprint 1

- 19/10/14
 - Conception / modélisation
 - Cahier des charges
 - Maquettes de l'interface
 - Cas d'utilisations

Post-sprint 1

- 24/10/14 - Application minimale pour validation des choix architecturaux - L'application devra permettre à l'utilisateur de pouvoir s'identifier et de visualiser et déposer des sujets - Mise en place de la base de données, LDAP - CAS, interface, rôles (étudiant, enseignant, admin), mise en place MVC avec AngularJS et Symfony
- 26/10/14 - Bilan du sprint 1 (tests, fonctionnalités, retour sur technologies, ...)

Post-sprint 2

- 28/11/14 - Application quasi-fonctionnelle - L'application devra permettre de gérer les candidatures des étudiants à un sujet avec la validation de la candidature par l'enseignant encadrant et effectuer la notation des étudiants, et éventuellement un début de la planification des soutenances
- 30/11/14 - Bilan du sprint 2

Post-sprint 3

- 30/01/15
 - Version finale/beta - L'application devra au mieux être achevée avec toutes les fonctionnalités décrites avant (cf. partie Objet et contexte)
 - Module d'interfaçage avec le SI de Lyon1
- 01/02/15 - Bilan du sprint 3
- 06/02/15
 - Documentation de développement
 - Documentation de déploiement
 - Documentation d'utilisation
 - Vidéo/Tutoriel de présentation de l'application
 - Rapport de retour sur les technologies utilisées

C- Autres livrables

- 10/02/15 - Support de présentation de soutenance

III- METHODE ET OUTILS

A- Contraintes

- **Contraintes de délais** : Ce projet doit arriver à minima en version Bêta dans les contraintes de temps de l'UE T15 Projet, et être applicable pour le projet de recherche de M1 l'année prochaine.
- **Contraintes de coûts** : Pour 5 personnes, 465h de travail sont prévues au total.
- **Contraintes de qualité** : Il s'agit d'une application légère et performante. Chaque année, il y aura environ 200 personnes qui utiliseront cette application. De plus, elle doit s'intégrer parfaitement avec le SI de l'UCBL (authentification CAS, LDAP). Dans le meilleur des cas, il faudrait pouvoir interfacier Spiral (recupération et dépôt des rapports) et TOMUSS (pour la saisie des notes) avec notre application.

B- Méthode

Au cours de la réalisation de ce projet, nous allons appliquer la méthode Agile « **Scrum** » pour la gestion du projet. La durée du projet est découpée en itérations courtes, nommés « Sprint ». Il y aura au total 3 sprints. Avant chaque sprint, les tâches sont définies pendant la réunion initiale : qui fait quoi, comment... et à la fin de chaque sprint, une réunion avec le client présente une démonstration de ce qui est fait, et prépare les différents points à aborder pour la suite.

C- Outils

a) Détails techniques :

Au niveau des détails techniques, après avoir étudié les différents choix possibles avec les contraintes mentionnées au-dessus, nous utiliserons durant ce projet les technologies suivantes :

- *Framework PHP côté serveur* : **Symfony2**. C'est un puissant framework qui nous permettra de réaliser le site de façon bien structurée.
- Grâce à une vaste librairie de « bundle » fournie par la communauté de **Symfony**, nous pourront implémenter judicieusement les besoins de ce projet comme l'utilisation de REST, la sécurité (l'authentification par CAS, le management des utilisateurs...).

- *Framework JavaScript côté client: **AngularJS***. Nous l'avons choisi en tant que framework qui peut faciliter la création des sites dynamiques et satisfaire les besoins d'une application SDI. Nous utilisons aussi **RequireJS** pour la gestion des modules JavaScript et l'organisation du code d'une manière plus efficace.
- *Front-end Framework: **Bootstrap***. Il s'agit d'un outil qui simplifie la création de l'architecture d'une page web grâce à différentes ressources préexistantes (feuilles CSS, jeu d'images, ...).
- *Système de gestion de base des données : **MySQL***. Pour ce projet, nous n'avons besoin que d'un SGBD léger et simple à utiliser. **MySQL** est donc un choix convenable.
- *Serveur de test : **Apache HTTP Server***. C'est un serveur HTTP qui peut prendre en charge plusieurs modules de PHP, le langage principal de notre projet côté serveur.

b) Gestion de projet :

Pour effectuer ce projet, nous utiliserons **la forge** de Lyon 1 avec laquelle nous devons utiliser les outils de communication pour mener à bien ce projet:

- La relecture de code
- Le gestionnaire de ticket
- La progression des tâches (heure, pourcentage, ...)
- Message de commit significatif avec référence au ticket assigné

IV- MACRO-PLANNING

A- Lots de travail

- Mise en place BDD

La base de données sera dédiée à l'application dans le sens où l'annuaire LDAP nous fournira déjà les informations sur les utilisateurs (donc éviter les redondances).

- Mise en place LDAP/CAS

La connexion à l'application se fera à partir du CAS et nous permettra d'identifier les utilisateurs et de définir leur rôle. Ce sera le point de départ pour le développement du projet et sera donc attaquée dès la première itération.

- Script serveur PHP

La partie serveur utilisera Symfony 2 et se comportera comme une API Rest.

- Client SPA

Le client sera une « Simple Page Application » et sera développée avec AngularJS. Il présentera 3 rôles : étudiant, enseignant et administrateur. Les fonctionnalités et les rôles seront développés en fonction de leur niveau d'importance dans les différentes itérations.

- Interfaces graphiques

Comme le client, on identifiera 3 rôles. Au lieu d'utiliser le moteur de Template d'AngularJS, nous utiliserons Bootstrap.

- Documentation

La documentation sera développée tout au long du projet et comprendra aussi un manuel pour l'utilisateur et pour le développeur.

- Tests

Les tests s'effectueront tous au long du projet. On discernera les tests unitaires (PHPUnit, JSUnit) et les bugs réparables lors de sessions de mise en situation.

- Retour sur les technos utilisées

Les outils utilisés tels que Symfony 2, AngularJS et Bootstrap sont des technologies assez récentes ou non présentes sur le système actuel de l'université. L'objectif de cette tâche est de prendre des notes sur leur utilisation et faire un « échange de connaissance » avec nos responsables de projet.

B- Phasage

Voici ci-dessous la répartition de nos différents lots de travail dans le temps :



	Itération 1	Itération 2	Itération 3
Pré-Sprints	S41-S42	S46-S47	S2-S4
Sprints	S43	S48	S5
Recettes	S44	S49	S6