

UNIVERSITÉ CLAUDE BERNARD

TER-PRIM : Application web de gestion des TER de l'UE MIF20

MASTER 2 INFORMATIQUE

---

Équipe de projet

BOUDEFFAR Abdelwadoud

BOUHAMDANI Florian

KACEM Imen

NGUYEN Le Duc Tan

NGUYEN Quoc Vuong

THAO John

---

Encadrants

PLANTEVIT Marc

COQUERY Emmanuel

---

Projet : <http://forge.univ-lyon1.fr/projects/p1307717-ter-prim>

Date : 15/10/2015

## Table des matières

1. [Objet et contexte](#)
  - a. [Contexte](#)
  - b. [Positionnement](#)
  - c. [Objet](#)
2. [Résultats attendus](#)
  - a. [Livrable de gestion de projet](#)
  - b. [Livrables techniques](#)
  - c. [Autres livrables](#)
3. [Méthodes et outils](#)
  - a. [Contraintes](#)
  - b. [Méthodes](#)
  - c. [Outils](#)
4. [Macro-planning](#)
  - a. [Lots de travail](#)
  - b. [Phasage](#)

Date	Révision
15 octobre	Corrections mineures
14 octobre	Corrections avec les remarques des encadrants
7 octobre 2015	Rédaction du dossier d'initialisation

# 1. Objet et contexte

## a. Contexte

Dans le cadre de l'unité d'enseignement (UE) du Master 1 Projet de Recherche (MIF20), le projet TER-PRIM a pour but de faciliter la gestion de celle-ci. Ce sujet a été proposé par M. PLANTEVIT. Le projet a déjà été développé par des étudiants de l'année universitaire 2014-2015. Notre objectif est de réaliser ce projet en utilisant de nouveaux outils et en nous appuyant sur l'existant.

Pour le moment, l'application se présente sous la forme d'un tableur TOMUSS qui liste les sujets proposés par les professeurs. Actuellement la matière se déroule comme suit :

- les enseignants déposent un ou plusieurs sujets de recherche sur MOTUS (application actuelle). L'enseignant précise le nombre d'étudiants acceptés pour un sujet;
- après avoir pris connaissance des sujets, les étudiants contactent les enseignants dans le but d'en avoir l'obtention;
- l'enseignant en question valide ou refuse la candidature et les étudiants notent leurs noms dans les cases correspondantes du tableur ;
- à la fin du projet, les étudiants se voient attribuer différentes notes à partir du rapport, de la soutenance et de l'appréciation de ou des encadrants.

## b. Positionnement

L'application actuelle de l'UE a de nombreuses contraintes qui provoquent une perte de temps et un problème de gestion pour le responsable. On constate les problèmes suivants :

- 3 outils différents pour gérer l'UE :
  - o MOTUS, pour déposer les sujets et inscrire les étudiants
  - o SPIRAL, pour le dépôt des différents livrables
  - o TOMUSS, pour la notation des étudiants
- Les étudiants s'affectent eux-mêmes sur un projet. Cependant, il est possible de le faire même sans l'accord de l'enseignant ce qui va à l'encontre du déroulement de l'UE.
- Il n'existe pas d'outils d'aide à la gestion du planning de soutenances. Il est fait à la main.
- La feuille TOMUSS pose un problème global d'ergonomie. En effet, la lecture des sujets et la navigation entre ces sujets sont particulièrement difficile. L'utilisation de filtres n'est pas assez mis en avant et donc pas assez utiliser.

## c. Objet

L'objectif de ce projet est de répondre aux problématiques citées précédemment. Pour cela, la nouvelle application de gestion des sujets de TER devra centraliser au maximum les outils. Elle répondra aux points suivants :

- gestion des comptes enseignants et étudiants via LDAP ;
- dépôt et gestion des sujets ;
- un système d'acceptation, de refus de candidatures ;
- recherche de sujets via un moteur ou par un système de filtres (sujet, enseignant, mots clés...) ;
- aide à la planification des soutenances ;
- gestion de l'évaluation des étudiants.

## 2. Résultats attendus

### a. Livrable de gestion de projet

Les livrables de gestion de projet seront à rendre à la fin de chaque sprint, soit aux date ci-dessous.

- [ 26/10/15 ] GdP Post-sprint #1
- [ 07/12/15 ] GdP Post-sprint #2
- [ 08/02/16 ] GdP Post-sprint #3

### b. Livrables techniques

#### *Post-sprint #1*

- 26/10/15 - Rendu des spécifications
  - o Modèle de la base de données
  - o Cas d'utilisations
- 26/10/15 - L'application devra permettre à l'utilisateur de s'identifier via le CAS ; Mise en place de la base de données, récupération des informations utilisateurs depuis LDAP, gestion des droits utilisateurs, gestion de sujets
- 26/10/15 - Remise du dossier post sprint D1

#### *Post-sprint #2*

- 07/12/15 - L'application devra implémenter le module de candidatures des utilisateurs et la gestion du planning de soutenance. De plus, un thème aux couleurs du département informatique sera mis en place.
- 07/12/15 - Remise du dossier post sprint D2

#### *Post-sprint #3*

- 08/02/16 - Implémentation de la recherche, de la gestion de notation
- 08/02/16 - **Produit logiciel:** Application finale
- 08/02/16 - Documentations de développement logiciel, de déploiement, d'utilisation du logiciel
- 08/02/16 - Remise du dossier post sprint D3

### c. Autres livrables

- 16/02/16 - Soutenance, dossier de synthèse final et affiche de promotion

## 3. Méthodes et outils

### a. Contraintes

Nous allons maintenant énumérer les différentes contraintes imposées. Il y a deux types de contraintes. Les contraintes de gestions et les contraintes techniques.

La première est liée à la gestion de projet, plus précisément aux délais imposés. Effectivement, l'application devra être fonctionnelle, l'objectif étant de l'utiliser lors de la rentrée 2016/2017.

La seconde contrainte est due aux coûts. Selon nos estimations, le nombre d'heures de travail total est d'environ 500 heures.

Ensuite, côté technique, il y a une contrainte de qualité. En effet, l'application sera utilisée par un nombre important de personnes. L'authentification devra impérativement utiliser les systèmes d'informations de l'UCBL, à savoir le CAS et LDAP.

Enfin, nous avons une contrainte au niveau du matériel à utiliser. Nous devons déployer l'application sur un serveur PHP avec une base de donnée MYSQL.

## b. Méthodes

Pour mener à bien ce projet, nous allons utiliser la méthode Agile "**Scrum**". Les temps de travail assignés à ce projet sont des itérations d'une semaine nommées "sprints". Les sprints se compteront au nombre de trois. Avant chacun d'eux, une réunion se tiendra pour définir attribuer les tâches à chaque membre de l'équipe. Cette méthode permet aussi d'organiser quotidiennement des échanges sur les différents problèmes rencontrés actuellement ou simplement faire un point sur l'avancée du projet. Ensuite, à chaque fin de sprint nous faisons un bilan afin de vérifier si nous avons respecté ou non notre planning et présentons des prototypes au client pendant la semaine de recette. Suite à quoi, il est possible d'adapter le planning en fonction de l'avance ou du retard accumulé.

## c. Outils

Pour la partie de gestion de projet nous utiliserons la Forge de l'université Claude Bernard. Cet outil nous permet de travailler en équipe, de partager des fichiers, gérer le versionnement de ces fichiers. Mais aussi la gestion des bogues grâce au module de tickets.

De plus, nous utiliserons l'application Trello pour la gestion des sprints et des tâches à réaliser. Elle nous permet d'attribuer des tâches à certaines personnes et de voir l'avancement du sprint en regardant les tâches à effectuer, en train d'être effectués ou complètement réalisées.

## d. Frameworks et bibliothèques

Pour mener à bien ce projet, nous allons utiliser différents frameworks. En effet, en ce qui concerne le développement de l'application, nous utiliserons différents outils. D'abord pour l'interface de l'application nous allons utiliser des technologies comme HTML, CSS, Bootstrap, AngularJS et RequireJS. Bootstrap est un framework permettant de faciliter le développement front-end et d'améliorer l'ergonomie des différentes pages. AngularJS est un framework javascript permettant de créer une single page application et de dynamiser nos pages web. Ce framework nous permettra de mieux gérer l'organisation de nos modules et de mettre en place rapidement une architecture Client/Serveur.

Ensuite, côté serveur, ou encore nommé back-end, nous allons utiliser le langage PHP avec le framework CodeIgniter. Nous allons aussi utiliser une base de donnée MYSQL et le langage SQL pour effectuer des requêtes. CodeIgniter nous permettra de gérer une API REST et d'effectuer du code métier en relation directe avec la base de données. Il va aussi nous permettre de nous connecter au système d'information de l'université grâce au CAS et au LDAP mis à disposition.

## 4. Macro-planning

### a. Lots de travail

Nous avons divisé le travail à effectuer en différents modules. Voici une liste et une description de ces modules.

- Mise en place de la base de données

La base de données servira principalement à stocker les sujets, les notes. En effet, l'annuaire LDAP fournira déjà les informations sur les utilisateurs.

- Connexion LDAP/CAS

Le CAS est utilisé pour proposer une connexion sécurisée et facile aux utilisateurs. LDAP permettra d'accéder aux informations des utilisateurs. Cela nous permettra aussi de gérer les droits des utilisateurs. Il existe trois types d'utilisateurs. Les étudiants, les professeurs et les administrateurs. Chaque type d'utilisateur à des droits différents.

- Module de gestion des sujets

Ce module aura pour but de gérer l'ajout, la modification, la suppression et listing de sujets.

- Module de recherche

Ce module permettra de faciliter les étudiants à trouver des projets qui leurs correspondent. Il sera possible lancer une recherche ou d'utiliser des filtres.

- Module de gestion des candidatures

Ce module permettra aux élèves de candidater à un seul sujet à la fois. L'enseignant pourra quant à lui accepter une ou plusieurs candidatures en fonction du nombre d'étudiants associés au projet. Les autres candidatures seront par conséquence annulées.

- Module de gestion des notes

Ce module aura pour but de gérer l'ajout, la modification, la suppression et le listing de notes. Il doit gérer les notes des rendus, des soutenances et l'exportation vers un fichier.

- Module de planification

Ce module aura pour but de faciliter la planification des soutenances. Elle devra permettre de planifier des soutenances pour chaque étudiant et attribuer un jury à chaque soutenance.

- Thème graphique

L'application reprendra le visuel du département Informatique pour une meilleure cohérence entre les services.

- Documentations

La documentation sera établie tout au long du projet pour faciliter la poursuite, l'utilisation et le déploiement de l'application. Elle fait parti des livrables techniques à rendre en fin de projet.

Pour obtenir plus de détails sur ces modules et connaître les tâches à effectuer, merci de se référer aux différents dossiers post sprint où les lots de travail seront expliqués plus en détail.

## b. Phasage

Voici la planification des tâches que nous avons effectuée pour ce projet. Il respecte les trois semaines de sprint réservées pour ce projet.

### Premier sprint (19/10/2015 - 24/10/2015)

- Mise en place de la base de données - 3 jours
- Connexion LDAP/CAS - 10 jours
- Module de gestion des sujets - 9 jours
- Dossier de fin de sprint - 2 jours

### Deuxième sprint (30/11/2015 - 04/12/2015)

- Module de gestion des candidatures - 3 jours
- Module de planification - 14 jours
- Thème graphique - 5 jours
- Dossier de fin de sprint - 2 jours

### Troisième sprint (01/02/2016 - 05/02/2016)

- Module de gestion des notes - 5 jours
- Module de recherche - 6 jours
- Documentations - 6 jours
- Dossier de fin de sprint - 2 jours

Il faut savoir que les nombres de jours indiqués sont des prévisions. Pour un sprint d'une semaine il y a 4 jours de travail par personne donc environ 24 jours au total de l'équipe. La prévision est un total par rapport au nombre de jours maximum de l'équipe. Par exemple pour la connexion LDAP/CAS, nous prévoyons 10 jours, mais deux personnes sur ce lot de travail. Ce qui fait 5 jours par personne.

Pour ce troisième sprint nous laissons des jours libres. Ils permettront de régler certains bogues de l'application, pour différents tests, pour ajouter des tâches précédentes non finies etc.