



HAL
open science

Mise en oeuvre des évolutions de LabBook, plateforme collaborative d'apprentissage pour les sciences expérimentales

Arnaud Le Meillour

► To cite this version:

Arnaud Le Meillour. Mise en oeuvre des évolutions de LabBook, plateforme collaborative d'apprentissage pour les sciences expérimentales. Environnements Informatiques pour l'Apprentissage Humain. 2016. dumas-01697755

HAL Id: dumas-01697755

<https://dumas.ccsd.cnrs.fr/dumas-01697755v1>

Submitted on 31 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL RHÔNE-ALPES

MÉMOIRE

Présenté par **Arnaud LE MEILLOUR**

En vue d'obtenir

LE DIPLÔME D'INGENIEUR C.N.A.M.

En Informatique option Systèmes d'Information (ISI)

Mise en œuvre des évolutions de LabBook,
plateforme collaborative d'apprentissage
pour les sciences expérimentales.

Soutenu le 25/01/2016

JURY

Président :	M. Éric GRESSIER SOUDAN	CNAM Paris
Membres :	M. Cédric D'HAM	LIG / MeTAH
	M. Marc FAYOLLE	CNAM Grenoble
	M. Philippe GARRAUD	CNAM Grenoble
	M^{me} Claire WAJEMAN	LIG / MeTAH

REMERCIEMENTS

Mes premiers remerciements s'adressent à mon institut, l'ESRF pour m'avoir accompagné et financé dans ma démarche de formation au long de ces années.

Je remercie le Fongecif Rhône-Alpes, sans lequel je n'aurais pu vivre cette enrichissante expérience qu'a été ce stage de 9 mois dans l'équipe MeTAH.

Je souhaite remercier la grande institution qu'est le CNAM et son personnel, pour m'avoir donné la chance de suivre des études supérieures à distance, parallèlement à la vie active.

Je remercie Monsieur Fayolle, mon tuteur de mémoire pour sa visite durant mon stage, et pour s'être porté garant de mon projet jusqu'à la soutenance.

Je remercie également tous les membres de ce jury pour leur présence, me faisant l'honneur d'évaluer mon travail.

Je remercie particulièrement mon tuteur de stage Cédric D'HAM pour m'avoir offert l'opportunité de réaliser un projet que j'avais à cœur, qui était de travailler dans le domaine du développement logiciel.

Un grand merci à l'équipe MeTAH, notamment à Nadine, Patricia, Claire, Isabelle, Patrick, Ben, Catherine et Reinaldo, pour m'avoir accueilli chaleureusement dans leur environnement.

Je remercie vivement mon collègue de bureau Emmanuel LÉTONDOR, qui poursuit le même cursus que le mien. J'ai bénéficié de ses conseils avisés et plus globalement de riches moments et échanges, dus à ses qualités autant humaines que techniques.

Je dédie ce mémoire à mes enfants Thomas, Lucile et Simon ainsi qu'à ma compagne et meilleure amie Valérie SCIBETTA qui m'a encouragé et aidé sans mesure.

LISTE DES ABREVIATIONS

AJAX	A synchronous J avaScript A nd X ML
CNRS	C entre N ational de la R echerche S cientifique
COPEX	C Oncevoir des P rotocols E Xpérimentaux
CSS	C ascading S tyle S heet
DOM	D ocument O bject M odel
EDI	E nvironnement de D éveloppement I ntégré
EER	E nhanced E ntity R elationship
EIAH	E nvironnements I nformatiques pour l' A pprentissage H umain
HTML	H ypertext M arkup L anguage
HTTP	H yper T ext T ransfert P rotocol
IDE	I ntegrated D evelopment E nvironment
IHM	I nterface H omme M achine
IMAG	I nstitut d' I nformatique et M athématiques A ppliquées de G renoble
INP	I nstitut P olytechnique de G renoble
INRIA	I nstitut N ational de R echerche en I nformatique et en A utomatique
ITS	I ntelligent T utoring S ystem
JSON	J avaScript O bject N otation
LIG	L aboratoire I nformatique de G renoble
MeTAH	M odèles et T echnologies pour l' A pprentissage H umain
MVC	M odèle- V ue- C ontrôleur
PDF	P ortable D ocument F ormat
PDO	P HP D ata O bject
PHP	H ypertext P reprocessor
SGBD	S ystème de G estion de B ase de D onnées
SQL	S tructured Q uery L anguage
SVN	S ubversion
UJF	U niversité J oseph- F ournier
UML	U nified M odeling L anguage
UPMF	U niversité P ierre- M endès- F rance

URL	Uniform R esource L ocator
WYSIWYG	W hat Y ou S ee I s W hat Y ou G et (ce que vous voyez est ce que vous obtenez)
XML	e X tensible M arkup L anguage
XSL	e X tensible S tylesheet L anguage

GLOSSAIRE

Balise :

Unité syntaxique délimitant des séquences de caractères à l'intérieur d'un flux de caractères.

Cadriciel :

Ensemble cohérent de composants logiciels à caractère générique qui guide l'architecture logicielle.

Patrons de conception :

Procédés de conception généraux décrivant des solutions standard en réponse à des problèmes de conception.

JQuery :

Bibliothèque JavaScript libre et multiplateforme facilitant l'écriture de scripts.

XML :

Langage de structuration de données à l'aide de balises.

XSL :

Langage de transformation qui permet de parcourir un document XML et de créer un fichier en sortie, XML ou non.

TABLE DES MATIÈRES

REMERCIEMENTS	1
LISTE DES ABREVIATIONS	2
GLOSSAIRE	3
TABLE DES MATIÈRES	4
INTRODUCTION	7
PREMIERE PARTIE - PROBLEMATIQUES FONCTIONNELLES	9
I. LE CONTEXTE	9
A. LE LABORATOIRE	9
B. L'EQUIPE	9
C. L'APPLICATION	10
D. L'HISTORIQUE	10
E. LE PROJET	11
II. ANALYSE DE L'EXISTANT	12
A. LABBOOK POUR L'ETUDIANT	12
B. L'INTERFACE ENSEIGNANTE INITIALE	20
C. BILAN DE L'EXISTANT	23
III. LES REALISATIONS	24
A. ERGONOMIE GENERALE	24
B. GESTION DES CLASSES ET DE LEURS ETUDIANTS	26
C. GESTION DES GROUPES	27
D. GESTION DES MISSIONS	28
E. SUIVIS DES RAPPORTS	29
F. AJOUT DE LA FONCTIONNALITE D'EDITION DES MISSIONS	30
IV. CONCLUSION DE LA PREMIERE PARTIE	35
DEUXIEME PARTIE - PROBLEMATIQUES TECHNIQUES	37
I. ARCHITECTURE DE LABBOOK	37
A. ARTICULATION INITIALE DES PAGES	38
B. METHODE INSPIREE DE MVC	38
C. GESTION DES COMPOSANTS MODULAIRES	40
D. CONCLUSION	41
II. LA BASE DE DONNEES	41
A. MISE A JOUR DES METHODES D'ACCES	43
B. SECURISATION DES ACCES	44
C. CONCLUSION	45
III. LE CSS ET LES CONVENTIONS DE NOMMAGE	46
A. LE NOMMAGE DANS L'ARCHITECTURE	46
B. LE CSS (CASCADING STYLE SHEETS)	47
C. LES METHODES D'UTILISATION DU CSS	48
D. LES PROBLEMATIQUES RENCONTREES	50
E. CONCLUSION	51
IV. ARCHITECTURE « AUTHORIZING »	52
A. EXIGENCES ET CONTRAINTES	52

B.	CHOIX DE L'ARCHITECTURE LOGICIELLE	53
C.	CONCEPTION DE LA PAGE D'EDITION DES MISSIONS.....	54
D.	CONCEPTION DES PAGES ENSEIGNANTES	55
E.	CONCLUSION	56
V.	CONCLUSION DE LA DEUXIEME PARTIE.	57
	TROISIEME PARTIE - PROBLEMATIQUES ORGANISATIONNELLES	59
I.	LES OUTILS UTILISES	59
A.	LE MAQUETTAGE	59
B.	LA PROGRAMMATION	60
C.	LA MODELISATION.....	62
D.	CONCLUSION	63
II.	LA GESTION DU PROJET	63
A.	QUALITES, COUT ET DELAIS (QCD)	63
B.	LES PERSONNES ET LES ROLES	63
C.	LE PERIMETRE DU PROJET ET LES LIVRABLES ATTENDUS	64
D.	LE PLANNING ET LE PILOTAGE	65
E.	CONCLUSION	66
III.	LES PRINCIPALES ETAPES	66
A.	LA RETRO-CONCEPTION.....	67
B.	L'AUTO-FORMATION	67
C.	LA PROGRAMMATION	68
D.	LA REINGENIERIE LOGICIELLE	68
E.	CONCLUSION	69
IV.	LA GESTION DE L'INFORMATION.....	69
A.	LA COMMUNICATION	69
B.	LES DOCUMENTS DE DEFINITION.....	70
A.	LES DOCUMENTS DE REALISATION ET DE SUIVI.....	71
B.	LES RAPPORTS	73
C.	CONCLUSION	73
V.	CONCLUSION DE LA TROISIEME PARTIE.	74
	CONCLUSION	75
A.	LE BILAN DU PROJET.....	75
B.	PERSPECTIVES ET EVOLUTIONS.....	75
C.	LE BILAN PERSONNEL	76
	LISTE DES FIGURES	78
	LISTE DES TABLEAUX	79
	LISTE DES EXTRAITS DE CODE.....	79
	BIBLIOGRAPHIE.....	80
	ANNEXES	82
D.	DIAGRAMME DE GANTT.	82
E.	DIAGRAMME DE LA BASE DE DONNEES	83
F.	SCHEMAS DE LA STRUCTURE DE FICHIERS DES PAGES WEB.	85
G.	SCHEMA DE LA SEQUENCE DES APPELS ENTRE FONCTIONS INITIEE PAR <i>GETANDDISPLAYLD()</i>	88
I.	DOCUMENT (PARTIEL) D'ENREGISTREMENT DES TESTS	89
K.	DOCUMENT (PARTIEL) DE SPECIFICATION DE LA PAGE <i>EDIT_MISSION.PHP</i>	97

INTRODUCTION

Le mémoire d'ingénieur et sa soutenance sont la formalisation d'un ensemble de compétences requises pour finaliser le long cursus d'ingénieur diplômé du CNAM. Nous devons lors de ce parcours de formation, montrer notre persévérance, notre curiosité, notre capacité à intégrer de nouvelles connaissances. Le stage final est le moment particulier où tous ces enseignements sont éprouvés dans le cadre d'un projet concret.

Ce stage a été l'occasion pour moi de montrer mes capacités dans plusieurs domaines. Le premier défi à relever était de devenir un développeur logiciel alors que je travaille dans l'informatique système depuis 15 ans. Je n'avais pas spécialement préparé ce virage par des formations spécifiques dans mon cursus au CNAM. J'ai dû, pour cette tâche et l'ensemble de mon stage, me montrer particulièrement autonome. J'ai débuté mon stage avec cette contradiction qui était d'être novice dans un domaine où je devais produire un travail d'ingénieur. Il m'a fallu acquérir le bagage technique nécessaire pour appréhender les demandes et besoins qui m'étaient formulés. J'ai utilisé les connaissances et compétences accumulées durant mon parcours professionnel et ma formation au CNAM pour aborder ces problématiques et les traiter comme on l'attend d'un ingénieur.

Ce stage s'est déroulé au Laboratoire Informatique de Grenoble (LIG) que je présenterai dans le chapitre de la première partie dédiée au contexte du projet. Cette première partie traite des aspects fonctionnels et métiers de l'application LabBook. J'y présenterai aussi l'interface pour les étudiants puis celle pour les enseignants dans son état initial. Après avoir pris connaissance de l'application, le chapitre suivant détaillera ma mission principale qui a été la refonte de l'interface des enseignants.

La deuxième partie de ce mémoire revient sur l'application LabBook avec une perspective technique. En premier lieu, je présenterai l'architecture globale et la base de données, pour venir ensuite sur la gestion des styles, le nommage et enfin une description plus détaillée des éléments mis en place.

La troisième et dernière partie est consacrée au stage dans ses autres caractéristiques telles que les outils logiciels utilisés et la gestion de projet. J'analyse ensuite les types d'activités principales menées durant ce stage et l'aspect de la documentation avant de conclure.

PREMIERE PARTIE - PROBLEMATIQUES FONCTIONNELLES

I. Le contexte

Dans ce chapitre, je présenterai le laboratoire et l'équipe en charge de la plateforme LabBook, avant d'en donner un bref descriptif. Je finirai par un rappel historique du projet pour mieux en comprendre le contexte.

A. Le laboratoire

Le Laboratoire d'Informatique de Grenoble (LIG) est issu du regroupement, le 1^{er} janvier 2007, de cinq anciens laboratoires de l'institut d'Informatique et Mathématiques Appliquées de Grenoble (IMAG) et du laboratoire de l'Institut National de Recherche en Informatique et en Automatique (INRIA) de Grenoble-Rhône-Alpes.

Le LIG est sous la tutelle conjointe de l'université Joseph-Fournier (UJF) de l'université Pierre-Mendès-France (UPMF), de l'institut Polytechnique de Grenoble (INP), du centre national de la recherche scientifique (CNRS) et est partenaire de l'INRIA.

Il rassemble près de 500 chercheurs, enseignants-chercheurs, doctorants et personnels en support à la recherche, déployés sur 22 équipes. Ces équipes travaillent dans cinq axes thématiques de recherche autour du projet scientifique principal l'"Informatique ambiante et durable" :

- Génie des Logiciels et des Systèmes d'Information.
- Méthodes Formelles, Modèles et Langages.
- **Systèmes Interactifs et Cognitifs.**
- Systèmes Répartis, Calcul Parallèle et Réseaux.
- Traitement de Données et de Connaissances à Grande Echelle.

B. L'équipe

L'équipe MeTAH (Modèles et Technologies pour l'Apprentissage Humain) travaille sur l'axe : Systèmes Interactifs et Cognitifs. Elle regroupe des informaticiens et didacticiens autour de la question de la conception, du développement et des usages des Environnements Informatiques pour l'Apprentissage Humain (EIAH).

MeTAH se donne pour objectif de comprendre comment les dimensions éducatives (didactiques ou pédagogiques) et les usages peuvent être pris en compte dans :

- la conception d'artéfacts informatiques techniques (micromondes, simulations, tuteurs intelligents, jeux pour l'apprentissage, environnements collaboratifs, etc.).
- la conception de descriptions calculables de leur utilisation (scénarios d'apprentissage, d'encadrement, etc.).
- la conception de modèles computationnels des connaissances épistémiques et didactiques et de fonctionnalités associées (mécanismes de rétroaction, supervision, etc.).

Cette équipe de 12 membres permanents est impliquée dans différents projets que sont :

- UnderTracks (plateforme de gestion de traces et de processus d'analyse).

- Cartographie des savoirs (conception et implémentation d'outils informatiques aidant à l'acquisition du socle commun de connaissances et de compétences).
- CSCL (Computer Supported Collaborative Learning).
- FORMID (FORMation Interactive à Distance).
- EDDBA (Exercices DataBase about Algorithmics).
- ROSE (Raisonnements Ontologiques pour les Systèmes d'enseignement).
- Copex-chimie (Concevoir un protocole expérimental en chimie analytique avec l'aide d'un tuteur artificiel et d'une simulation).
- **LabBook (rapports expérimentaux en ligne).**

C. L'application

LabBook est une **plateforme en ligne** pour la création collaborative de **rapports expérimentaux** en amont et/ou en aval de l'expérimentation. Elle a été conçue pour les travaux pratiques ou projets scientifiques, au cours desquels les étudiants doivent produire des rendus à destination de leur enseignant.

Les rapports sont élaborés par des groupes de travail qui bénéficient de **services de collaboration** permettant de travailler à distance ensemble et avec leurs enseignants.

La plateforme propose actuellement quatre **outils permettant de concevoir les LabDocs** qui composent les rapports.

Ces outils sont spécifiquement adaptés pour la production de rapports scientifiques dans un cadre d'enseignement. L'outil **Copex** permet de concevoir des protocoles expérimentaux qui seront testés en laboratoire. Le second outil **Fitex**, aide à analyser et traiter les résultats des travaux expérimentaux avec un tableur spécialement adapté. LabBook propose également l'outil **Drawex** pour l'élaboration d'illustrations graphiques, ainsi qu'un éditeur de texte formaté.

L'axe de recherche principal autour de LabBook est l'impact sur l'apprentissage de l'étayage de l'activité de conception expérimentale. Ceci au moyen d'un travail sur la structuration de la tâche et sur la production de rétroactions individualisées.

D. L'historique

Les projets gérés par l'équipe MeTAH sont des supports à différents axes de recherches. Ces projets se suivent et peuvent reprendre tout ou partie de projets antérieurs car les domaines de recherche sont souvent connexes. Dans cette logique, plusieurs projets ont participé en amont à ce qu'est LabBook aujourd'hui :

- **Le projet Educaffix** (2001-2004) a été développé en collaboration avec l'entreprise CCE. L'objectif était de proposer des manipulations de chimie, pilotées à distance (remote-lab) à travers internet.
- **Le projet Copex** (2005-2008) avait pour objectif de concevoir un environnement informatique d'apprentissage humain (EIAH) aidant les élèves dans l'activité de conception d'expérimentations en sciences expérimentales (biologie, chimie, physique et sciences de la terre).
- **Le projet SCY - Science Created by You** (2008-2011) avait pour objectif de développer un environnement d'apprentissage flexible permettant à des élèves de cycle 2 de s'engager activement dans la résolution de missions scientifiques. Des versions Java de Copex et Fitex ont été développées dans ce projet.

- **Le projet Copex-chimie** (en cours) a remplacé le robot du logiciel Educaffix par une simulation. Copex-chimie met l'accent sur l'élaboration du protocole expérimental par les apprenants. Le problème proposé est le dosage par spectrophotométrie du colorant E124 (colorant rouge) contenu dans un sirop de grenadine. Les apprenants conçoivent le protocole expérimental avec l'aide d'un tuteur artificiel qui évalue ce protocole à la demande. Ils testent ensuite leur protocole en demandant des retours simulés sur la base de leur production.

De multiples stagiaires ont participé à la réalisation de LabBook, je présente ci-dessous une chronologie de leurs interventions :

2008 : **Marjolaine BODIN**, a développé les versions java de Copex et Fitex au cours du projet SCY jusqu'en 2010. Son code a été progressivement remplacé pour s'affranchir de la technologie Java.

2010 : **Albin MANOEL** a travaillé sur les bases de LabBook.

2012 : **Morgane BERTAUX** a fortement fait avancer les aspects collaboratifs de LabBook avec les outils de ressources et de messagerie.

Florence BRESTAZ a initialisé la reprise de Copex en JavaScript.

2013 : **Heda WANG** a commencé l'étude sur l'outil de dessin.

Kamil KOLODZIEJCZYK a pris la suite de Heda et a proposé une première version de l'outil de dessin qui a été finalement redéveloppé par un prestataire extérieur.

Julien BOUCHON a développé la première version Fitex en JavaScript.

2014 : **Emmanuel CARRÉ** a fait évoluer de manière importante Copex et LabBook.

2015 : **Arnaud LE MEILLOUR**, refonte de l'interface auteur.

Alexandre GANNE continue le travail sur l'outil Fitex.

E. Le projet

Mon projet a consisté, dans sa majeure partie, en la refonte des interfaces de l'enseignant dans l'application LabBook. Des problèmes d'ergonomie avaient été signalés et certaines fonctionnalités dont la création de missions n'étaient pas disponibles.

J'ai étudié l'application dans son état initial pour en comprendre le fonctionnement et en faire un diagnostic. Avec ces informations et les recherches effectuées sur internet, j'ai proposé et mis en œuvre cette nouvelle interface.

Le projet ne bénéficie pas de budget, ce qui explique l'utilisation de technologies gratuites, et d'un développement par le biais de stagiaires.

Ce stage n'était pas encadré dans un service informatique, ce qui explique mes démarches de recherches personnelles.

II. Analyse de l'existant

LabBook est utilisé par deux types d'utilisateurs distincts que sont les enseignants et les étudiants.

- **L'enseignant** crée des missions qui se composent d'éléments d'information et d'une structure de rapports destinés à être complétés par les étudiants. Il gère des classes d'étudiants et affecte les missions à des groupes d'étudiants. Il va ensuite suivre la rédaction des rapports et commenter leurs contenus.
- **L'étudiant** accède aux missions qui lui sont affectées et complète le rapport lié à celle-ci au moyen de différents outils. Il travaille en groupe avec lequel il peut échanger par le biais d'une messagerie. Une fois le rapport complété par le groupe, il est restitué à l'enseignant.

Je vais détailler de manière plus précise ces deux rôles et les interfaces associées.

A. LabBook pour l'étudiant

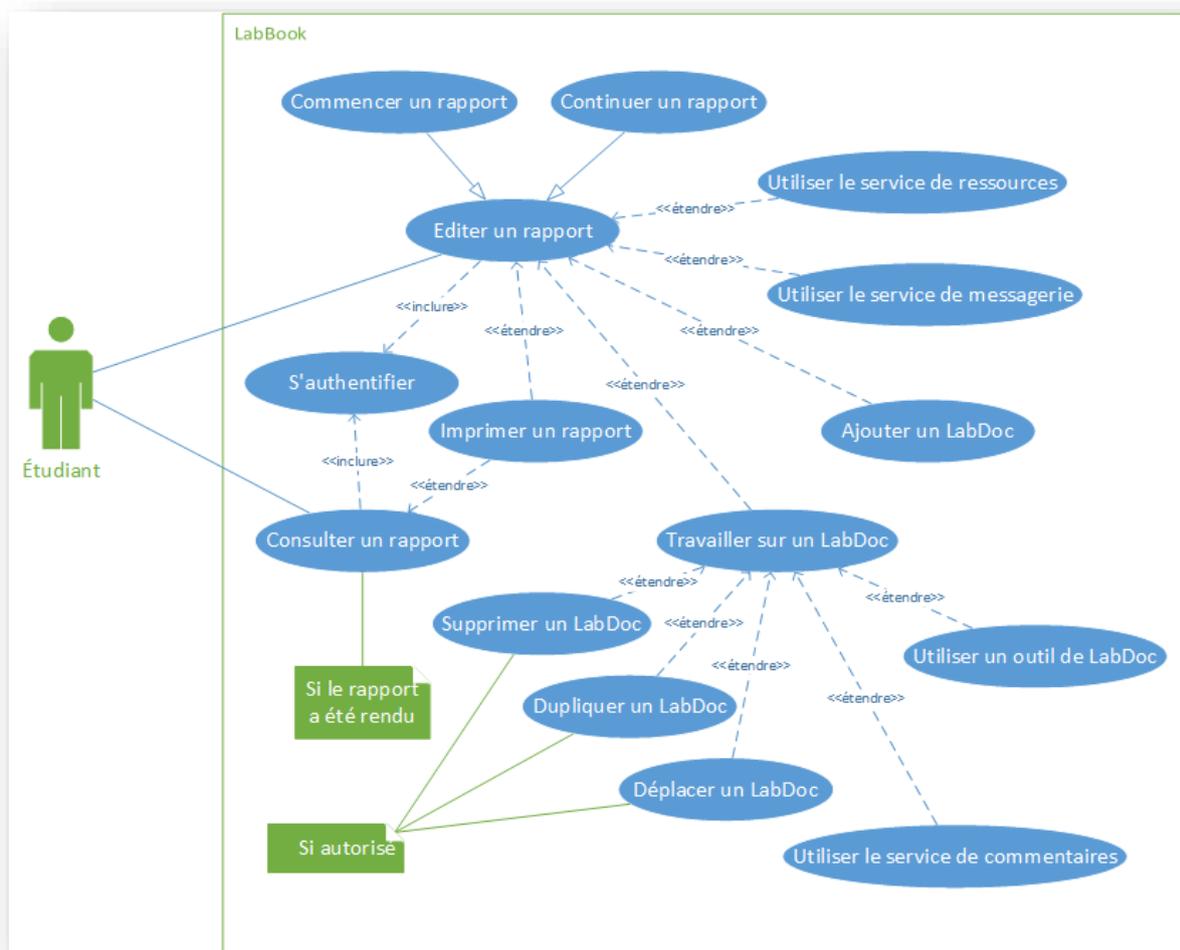


Figure 1 : Cas d'utilisation de LabBook pour un étudiant.

Les étudiants travaillent par groupe sur l'élaboration d'un rapport qui doit être rendu à leur enseignant.

Un des enjeux de l'application est le travail **collaboratif** entre les différents acteurs. Les élèves et l'enseignant peuvent communiquer par le biais de services intégrés de commentaires et de messagerie.

L'application doit gérer le travail **multi-utilisateurs** sur les rapports avec un système de verrouillage sur des éléments actifs.

LabBook est aussi une **plateforme distante**, ce qui implique un travail particulier autour de la validité et de la sécurité des données.

Pour mieux comprendre le fonctionnement de LabBook pour un étudiant, il est préférable d'en décrire les composants.

1. L'accès aux missions

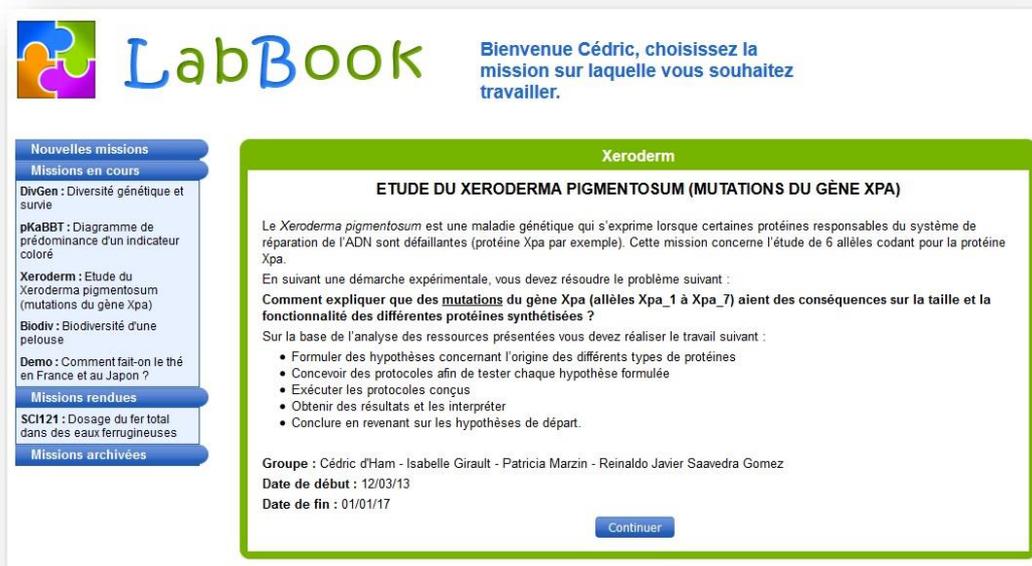


Figure 2 : Interface de choix des missions de l'étudiant.

L'étudiant, après s'être authentifié, accède à son espace de travail (cf. Figure 2). Selon que les rapports des missions ont été rendus ou pas, l'étudiant a la possibilité de travailler dessus ou seulement de les consulter.

L'écran se compose d'un menu latéral gauche proposant les missions de l'étudiant, dans leurs quatre états possibles:

- **Nouvelles missions** (missions non-débutées).
- **Missions en cours** (au moins un étudiant est entré dans l'espace de travail de la mission).
- **Missions rendues** (le rapport a été soumis à l'enseignant par un étudiant).
- **Missions archivées** (le rapport est considéré comme terminé par l'enseignant).

Un panneau central donne les éléments descriptifs de la mission et permet d'y accéder dans le mode autorisé (consultation/édition) par un bouton. On retrouvera à cet endroit le nom des membres du groupe ainsi que les dates de début et de fin d'édition du rapport.

Le rapport et le groupe auquel il est lié sont des notions similaires dans LabBook car un groupe est lié à un unique rapport.

Le rapport est construit dans le contexte d'une mission préalablement créée par un enseignant. L'étudiant voit dans son interface ses missions, qu'il peut continuer. L'interface d'édition du rapport ne mentionne pas le terme de rapport directement, mais la mission sur laquelle l'étudiant travaille et la possibilité d'en changer dans le menu. Dans le menu, le terme de rapport est explicitement utilisé lors des opérations de restitution à l'enseignant ou d'impression.

2. Interface d'édition du rapport



Figure 3 : Interface d'édition de la mission étudiante.

L'interface principale de l'étudiant est celle où il consulte les différents éléments de la mission et construit son rapport (cf. Figure 3). Un menu supérieur propose l'accès à différentes actions :

- Affichage des informations de l'étudiant.
- Retour à la page de choix des missions.
- Restitution du rapport à l'enseignant.
- Déconnexion.
- Ouverture du service de ressources.
- Ouverture du service de messagerie.

- Impression de tout ou partie du rapport.

Les membres du groupe disposent d'un **rapport pré-structuré en parties**. Chaque partie est accompagnée optionnellement d'une **consigne** et de **LabDocs** mis à disposition par l'enseignant concepteur de la mission. Ces LabDocs sont des documents créés dans l'application LabBook au moyen de différents outils détaillés dans la suite de ce chapitre.

À partir de ces éléments, les étudiants élaborent leur rapport, ils peuvent également créer leurs propres LabDocs.

3. Le service de ressources

Il donne accès aux consignes et documents que l'enseignant met à disposition du groupe pour élaborer son rapport (cf. Figure 4).

L'élève peut lui aussi enrichir les ressources avec ses propres documents, dans une partie qui lui est consacrée et ceux-ci seront disponibles pour tous les élèves du groupe.

Les documents sont soit des liens internet, soit des fichiers de type *PDF* ou encore des images.



Figure 4 : Interface du service de ressources.

4. Le service de messagerie

Il permet d'échanger des messages ainsi que des LabDocs entre élèves et à destination de leurs enseignants (cf. Figure 5). Le transfert du message n'est pas complètement instantané, un rafraîchissement est effectué toutes les 30 secondes.



Figure 5 : Interface du service de messagerie.

Des options de communications sont affectées au groupe lors de sa création :

- Pas d'accès au service de messagerie.
- Envoi de messages au sein du groupe uniquement.
- Envoi de messages au sein de la classe uniquement.
- Envoi de messages aux élèves de toutes les classes.
- Envoi de messages aux participants de cette mission uniquement.
- Envoi de messages pour toutes les missions.
- Possibilité d'envoyer des LabDocs.
- Possibilité d'enregistrer dans le rapport les LabDocs reçus.

5. Le LabDoc

Par défaut, dans l'interface étudiante, un LabDoc est présenté en « Lecture » et possède un bandeau supérieur avec plusieurs composants (cf. Figure 6) :

- Une icône (optionnelle) qui informe qu'un autre étudiant a effectué une modification non encore visualisée par l'étudiant connecté.
- Une icône (optionnelle) précisant que le LabDoc est verrouillé et ne peut être édité car il est ouvert par un autre étudiant du groupe.
- Une icône précisant le type de LabDoc.
- Le nom du LabDoc, modifiable quand le LabDoc est en édition.
- Le statut du LabDoc [Visible | Brouillon | En modification].
- L'icône de mise en édition du LabDoc.
- Les icônes de duplication, déplacement et suppression du LabDoc.
- L'icône d'accès au service de commentaire du LabDoc.



Figure 6 : Menu de gestion des LabDocs.

6. Les outils de création de LabDoc

L'enseignant définit pour chaque partie de rapport le type de LabDocs que l'étudiant pourra créer, dans les paramètres de la mission sur laquelle est basé le rapport. Les différents outils de création de LabDocs sont :

- Un éditeur de textes *WYSIWYG*.
- Drawex, un outil de dessins libres.
- Copex, pour la création de protocoles d'expérimentation.
- Fitex, outil de traitement et de représentation des données (graphes XY).

a) *L'outil de texte*

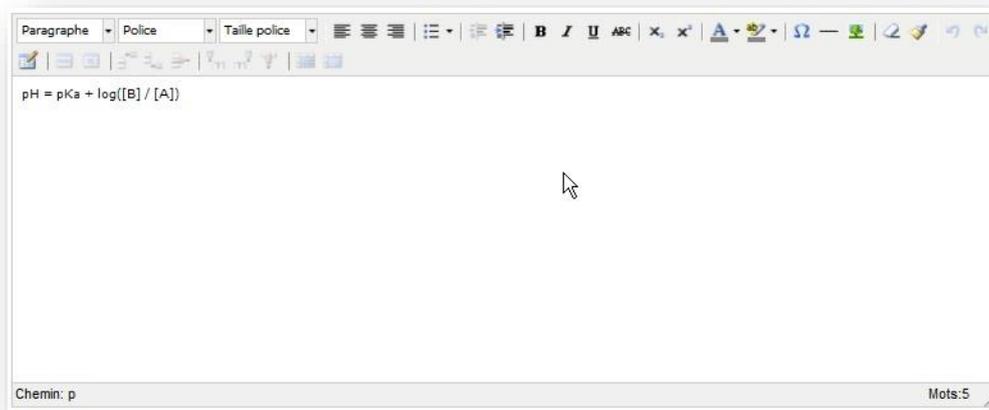


Figure 7 : Outil LabDoc de texte TinyMCE.

L'outil de LabDoc texte est TinyMCE [tinymce15]. Il s'agit d'un éditeur gratuit pour lequel une configuration personnalisée a été appliquée pour ne proposer que les options d'édition de texte souhaitées. L'étudiant peut l'utiliser pour intégrer des textes mis en forme et des images dans son rapport.

b) *L'outil de dessin*

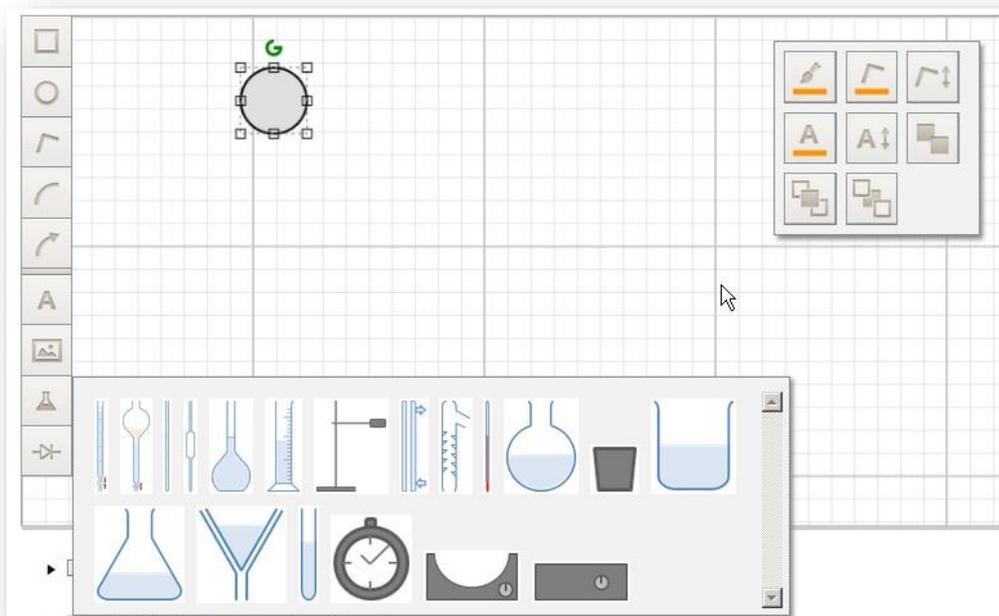


Figure 8 : Outil de LabDoc dessin Drawex.

L'outil de LabDoc dessin est appelé **Drawex**. Il s'agit ici du composant logiciel *Zwibbler* [Zwibbler15], commandé par le laboratoire à un prestataire extérieur. Il permet aux étudiants d'illustrer leur rapport avec des fonctionnalités de dessin standard ainsi que certaines images de composants scientifiques.

c) L'outil de protocole

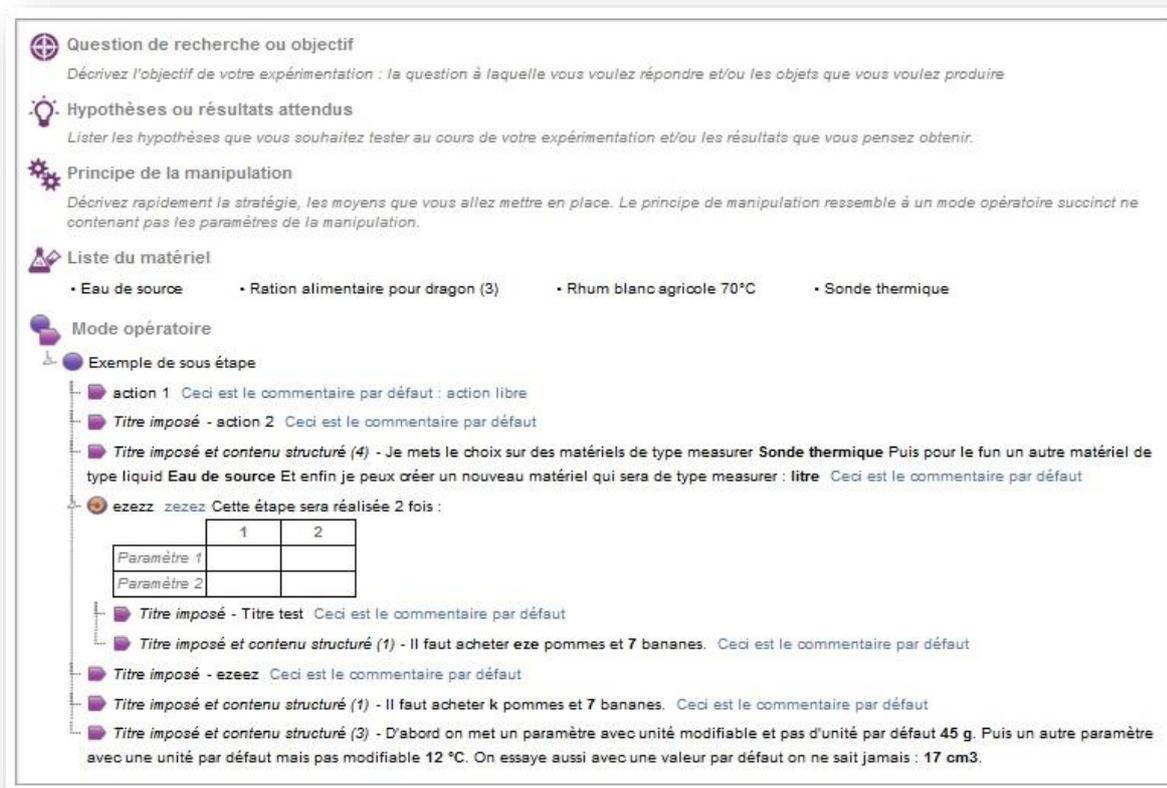


Figure 9 : Outil de LabDoc COPEX, pour la création de protocoles expérimentaux.

L'outil de LabDoc permettant la création de protocoles expérimentaux se nomme **Copex** (Concevoir des Protocoles Expérimentaux) (cf. Figure 9) et est issu d'un projet antérieur européen.

Un protocole se compose de plusieurs sections qui peuvent être complétées et structurées partiellement en amont par l'enseignant concepteur de la mission.

Le protocole expérimental comporte les sections suivantes :

- L'objectif expérimental visé ou la question à laquelle l'expérimentation tente de répondre.
- Les hypothèses proposées comme réponse à la question expérimentale, et qui sont testées au cours de l'expérimentation.
- Le principe général de la manipulation.
- La liste des matériels disponibles, ou créés par l'étudiant, avec leurs spécifications.
- La description de la manipulation faite au moyen d'une séquence d'actions et d'étapes structurées de manière hiérarchique.

d) L'outil de données

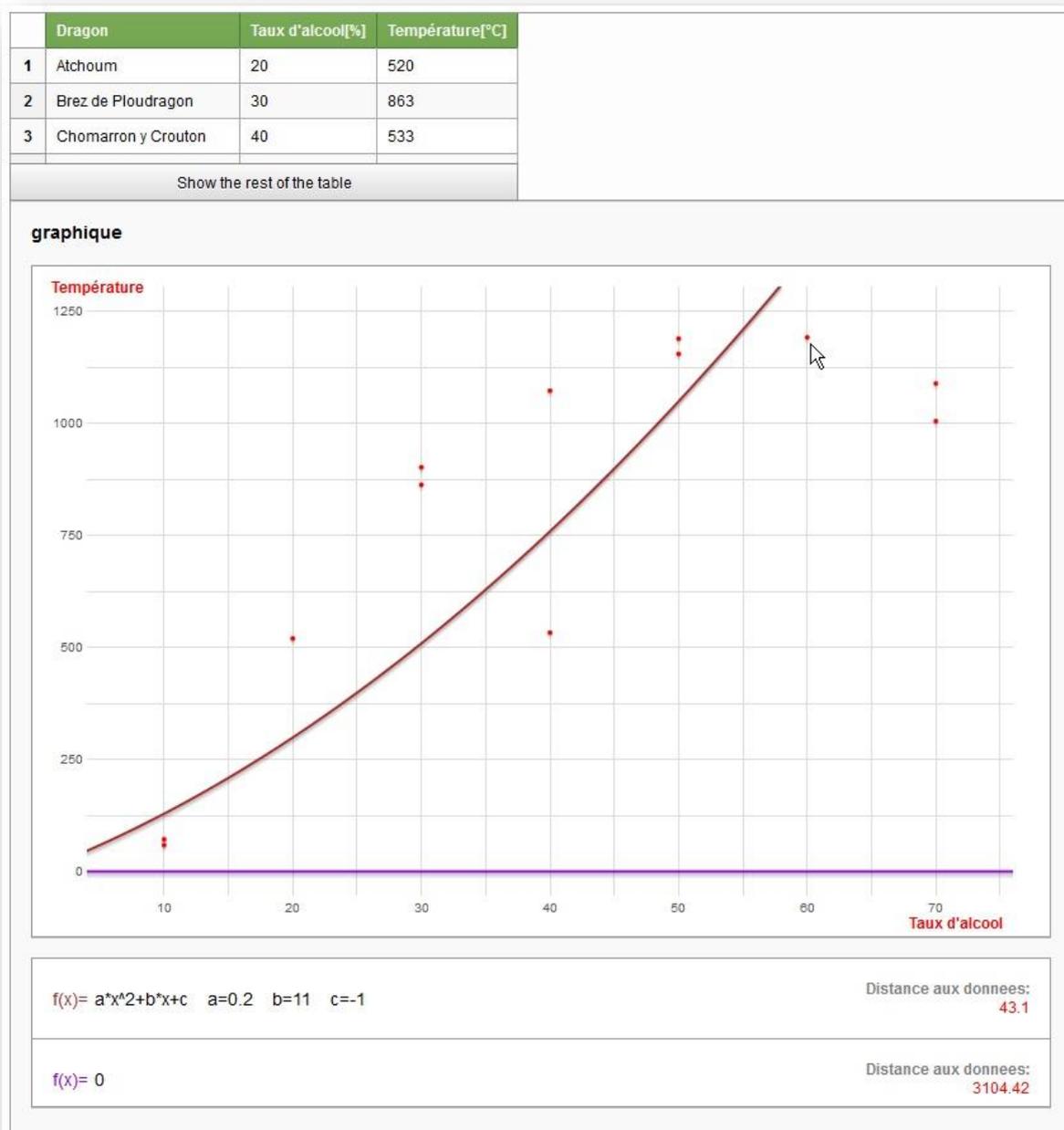


Figure 10 : Outil de LabDoc Fitex pour le traitement de données expérimentales.

Le dernier outil **Fitex** (cf. Figure 10), permet de gérer des données à la manière d'un tableur.

Chaque donnée est traitée en colonne avec la possibilité de choisir un nom, une unité et des paramètres d'affichage. Les fonctions classiques sont disponibles dans l'outil (somme, moyenne, tri...). Il y a également la possibilité d'utiliser des constantes qui sont soit prédéfinies, soit créées par l'utilisateur.

La représentation des données est possible sous la forme de graphes XY. Une des fonctionnalités intéressantes de Fitex est qu'il est possible d'ajuster manuellement les données par des modèles mathématiques sur le graphe XY.

L'ajustement manuel signifie que la fonction d'ajustement peut être définie avec des paramètres ; par exemple une fonction linéaire peut être définie par " $y = a*x + b$ ". Il est alors possible de modifier les valeurs des paramètres (a et b dans l'exemple) pour ajuster au plus près la fonction par rapport aux données expérimentales. Pour faciliter l'ajustement, Fitex calcule la distance entre la fonction et les données expérimentales [E. CARRÉ14].

B. L'interface enseignante initiale

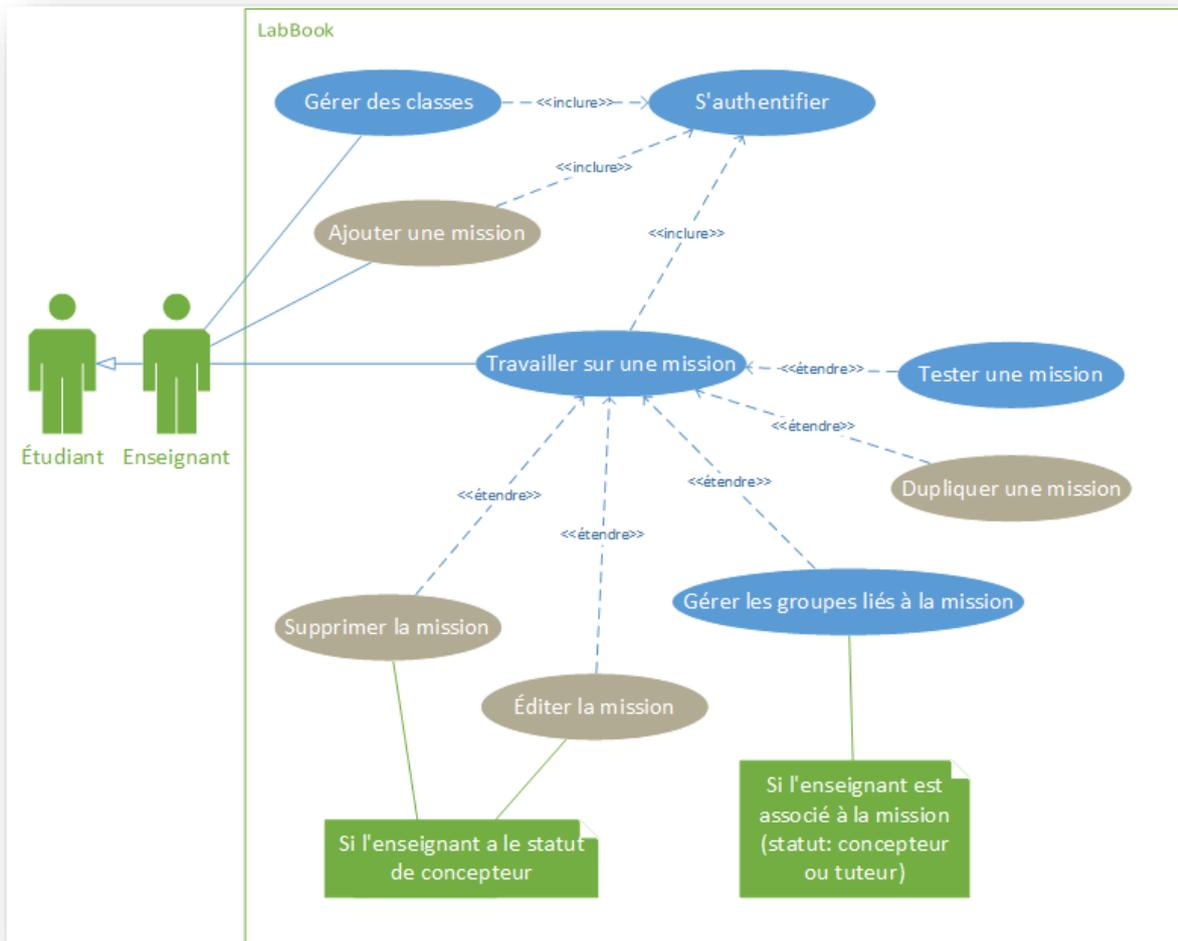


Figure 11 : Cas d'utilisation de LabBook par un enseignant.

L'enseignant dans LabBook bénéficie de toutes les prérogatives de l'étudiant auxquelles s'ajoutent les fonctionnalités qui lui sont propres.

Les enseignants peuvent aussi faire partie d'un groupe de travail lié à une mission et accéder aussi aux fonctionnalités en tant qu'étudiant.

1. La page de choix des missions

Après s'être authentifié, l'enseignant arrive sur la page de choix des missions (cf. Figure 12).

La page est celle d'un étudiant, à laquelle s'ajoute une barre de menu spécifique au-dessus de celle des étudiants. Ce menu propose les missions dont l'enseignant est **concepteur** (*missions créées*), celles dont il est **tuteur** (*missions gérées*) et les missions de statut *public* des autres enseignants, pour lesquelles il est **observateur** (*Autres missions*).

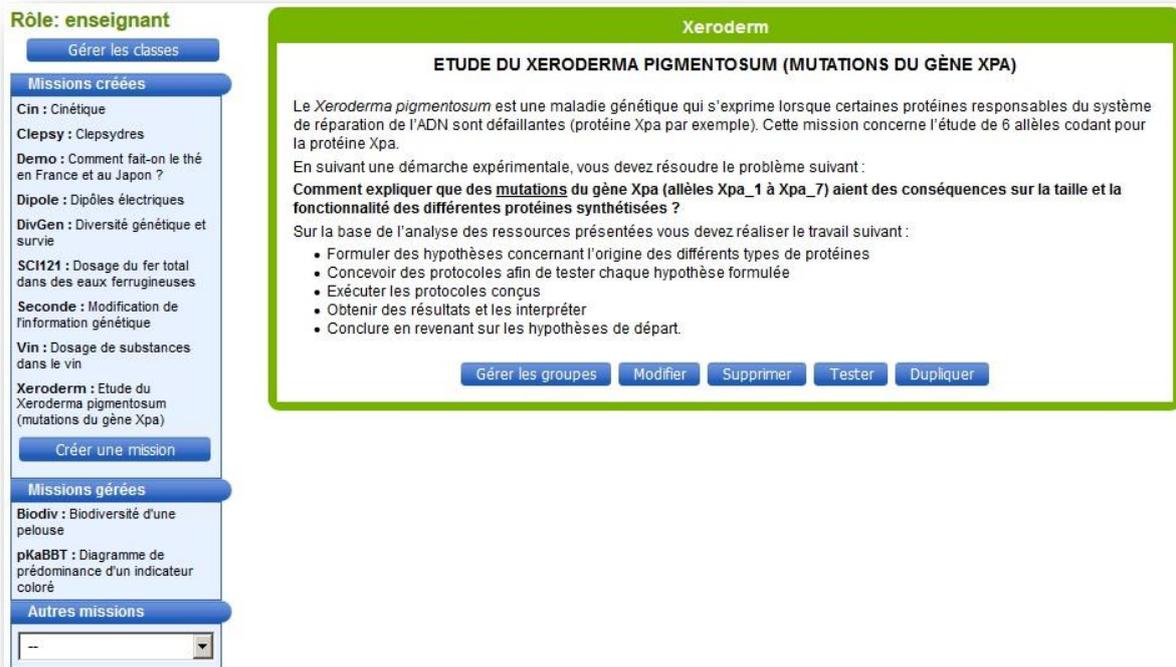


Figure 12 : Menu et boutons spécifiques aux enseignants (ancienne interface).

Toujours dans cette zone de menu, un bouton donne accès à la page de gestion des classes, le bouton de création de missions qui apparaît dans l'interface est non fonctionnel.

Le panneau descriptif de la mission se termine par des boutons réservés aux enseignants :

- **Gérer les groupes**, permet aussi le suivi des rapports.
- **Modifier** la mission, doit conduire à l'édition des missions (non fonctionnel).
- **Supprimer** la mission (non fonctionnel).
- **Tester** la mission (non fonctionnel).
- **Dupliquer** la mission (non fonctionnel).

Le clic sur le logo supérieur de LabBook renvoie sur la page hiérarchiquement précédente, qui est ici la page d'authentification.

2. Création de mission

L'enseignant crée une mission en fournissant des éléments d'informations, tels qu'une description ou des documents de ressources. Il construit également le squelette des rapports demandés aux étudiants par la définition de parties et de LabDocs. Il peut associer d'autres enseignants aux missions dont il est concepteur en leur donnant la qualité de tuteur ou de concepteur. Ce statut de concepteur, autorise l'enseignant à modifier les paramètres de la mission.

Au début de mon stage, cette fonctionnalité n'était pas disponible, et mon responsable devait intégrer les données directement dans la base de données.

3. La gestion des classes

Un enseignant peut créer des classes (cf. Figure 13) et y gérer ses élèves (ajouts, modifications, suppressions). L'importation d'élèves à partir d'une liste fournie n'était pas encore fonctionnelle.

Depuis cette interface, on peut aussi associer d'autres enseignants à ses classes, ce qui donne aux enseignants sélectionnés les mêmes prérogatives sur la classe. On peut voir également les missions attribuées aux étudiants et leur dernière connexion.

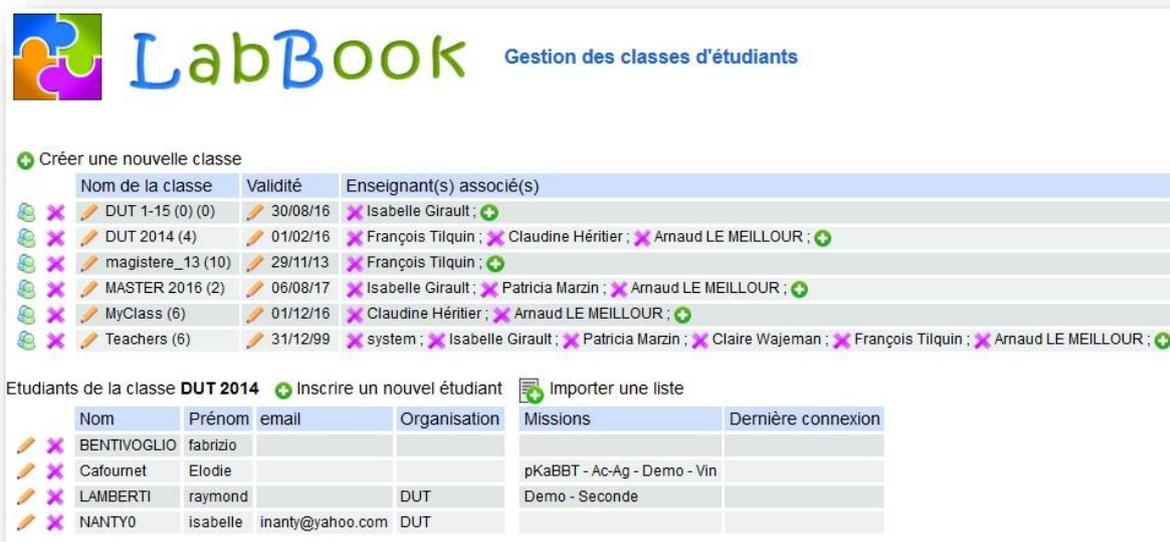


Figure 13 : Interfaces de gestion des classes (ancienne interface).

4. La gestion des groupes

Les enseignants associés à une mission (tuteurs ou concepteurs) ont la possibilité de créer des groupes d'étudiants associés à cette mission (cf. Figure 14). Quand on se réfère à un groupe, on identifie un rapport unique car un groupe n'est lié qu'à une seule mission.

L'interface de gestion des groupes est accessible via la mission sélectionnée au préalable dans l'interface principale de choix des missions. Elle permet le suivi de l'avancement des rapports des groupes liés à cette mission, en donnant leurs statuts (Non débuté, En cours, Rendu ou Archivé) ainsi que le nombre de LabDocs créés. L'enseignant peut aussi accéder depuis cette page au service de messagerie, et aux rapports en lecture pour les consulter et laisser des commentaires sur les LabDocs de son choix.

Les enseignants peuvent avoir des classes en commun auxquelles ils sont associés et d'autre non. Les groupes ne sont pas associés directement aux enseignants, mais par le biais de l'appartenance des élèves aux classes de l'enseignant (cf. Annexe E). Si un enseignant crée un groupe et y inclut des élèves de plusieurs classes, et que ces classes ont des enseignants associés différents, les enseignants non associés à ces classes auront des élèves « extérieurs » dans leurs groupes.

LabBOOK Gestion des groupes d'étudiants

Mission Demo : Comment fait-on le thé en France et au Japon ?
Designer(s) : **system, Cédric d'Ham** - Tuteur(s) :

Inscrire un nouveau groupe à cette mission

Ajouter un étudiant au groupe :
 Aucaigne Fanny ; BENTIVOGLIO fabrizio ; BLANC3 louis-emmanuel ; BLANC9 louis-emmanuel ; BOURDON3 ; BOURDONI ; Bourguignon Jeremie ; BRONNEC guillaume ; Buyse Pierre-Emmanuel ; Cafournet Elodie ; Cafournet Elodie ; CELMA stéfi3 ; CELMA3 stéfi ; Chazot Alban ; Chevrier Joël ; d Ham Cédric ; Detrait Maximin

Ajouter les étudiants d'un groupe existant :
 - : Dupouy Henri, Marzin Patricia, Raad Mariam, Saavedra Gomez Reinaldo Javier
 - : Cafournet Elodie, Chazot Alban, Chevrier Joel, Dufourd Thibault, Dupouy Henri, Girault Isabelle, JACQUI audrey, Saavedra Gomez Reinaldo, Javier Waieiman Claire, Zamorano Natalia

Etudiants du groupe : Bourguignon Jeremie, Cafournet Elodie

Nom du groupe (optionnel) : Grp01-15

Dates (optionnelles) de début [jj/mm/aa] et de fin [jj/mm/aa]

Envoi de messages : Au sein du groupe pour cette mission Envoi de LD

Enregistrement de LD

Groupe	Etudiants	Début	Fin	Avancement *	Dernière modif.	
<input checked="" type="checkbox"/> almtest45b	Pierre-Emmanuel Buyse, Elodie Cafournet, audrey JACQUI, Natalia Zamorano	31/07/15	02/03/16	Non débuté		
<input checked="" type="checkbox"/> almtest48	Thibault Dufourd, jean ROMAIN, raymond LAMBERTI	31/07/15		Non débuté		

* Avancement du rapport :
 Non débuté : la mission n'a pas été consultée.
 En cours : les étudiants ont accédé au moins une fois au rapport.
 Rendu : les étudiants attendent une validation de leur rapport par l'enseignant. Ils ne peuvent plus le modifier.
 Archivé : le rapport est terminé et non modifiable.

Figure 14 : Interface de gestion d'un groupe (ancienne interface).

C. Bilan de l'existant

Hormis les opérations sur les missions enseignantes, la plupart des fonctionnalités sont disponibles. Lors de mon analyse des interfaces existantes, j'ai relevé deux types de problèmes :

- **L'organisation des pages** pour l'enseignant : Dans sa page de choix des missions, l'enseignant cumule ses fonctionnalités avec celles des étudiants. Cela complexifie l'interface et selon le menu choisi, étudiant ou enseignant, le contexte change. L'enseignant a une vision orientée « mission », il ne peut suivre les groupes et leur travail de manière globale. Il doit passer par une mission.
- **L'ergonomie** au sein de certaines pages : Les pages de gestion des groupes et des classes présentent une ergonomie complètement différente de la page de « choix des missions ». Cette interface est fonctionnelle et synthétique par l'accès rapide à toutes les fonctionnalités, mais la multiplicité des icônes pour rendre ce service charge fortement la page visuellement. Cet effet est amplifié par les différentes couleurs utilisées.

III. Les réalisations

Les modifications présentées dans ce chapitre ne respectent pas l'ordre chronologique de création. Le premier travail a été la page d'édition des missions. Celle-ci était nouvelle et je ne suis parti d'aucune référence graphique de l'application. Le bloc constitué par les quatre pages qui font la base de l'interface de l'enseignant, a été développé dans un second temps. La page d'édition de mission a ensuite été retravaillée pour s'inscrire dans l'ensemble. Les travaux concernant d'autres pages, que ce soit dans un but d'optimisation ou de correction d'erreurs, ne seront pas traités dans cette partie réservée aux fonctionnalités et interfaces.

Plusieurs enseignants participent à ce projet en expérimentant LabBook dans des conditions réelles d'apprentissage. Ces personnes avaient fait part de leurs difficultés de prise en main de leur interface. Une partie des fonctionnalisés étaient déjà en place et l'enjeu était donc de repenser l'organisation et l'ergonomie des pages.

A. Ergonomie générale

Mon responsable m'a chargé de travailler sur des maquettes proposant une nouvelle interface en intégrant une présentation des données sous forme de tableaux. A partir de mes maquettes (cf. Figure 15), nous avons retravaillé avec mon responsable pour décider de la structure globale.

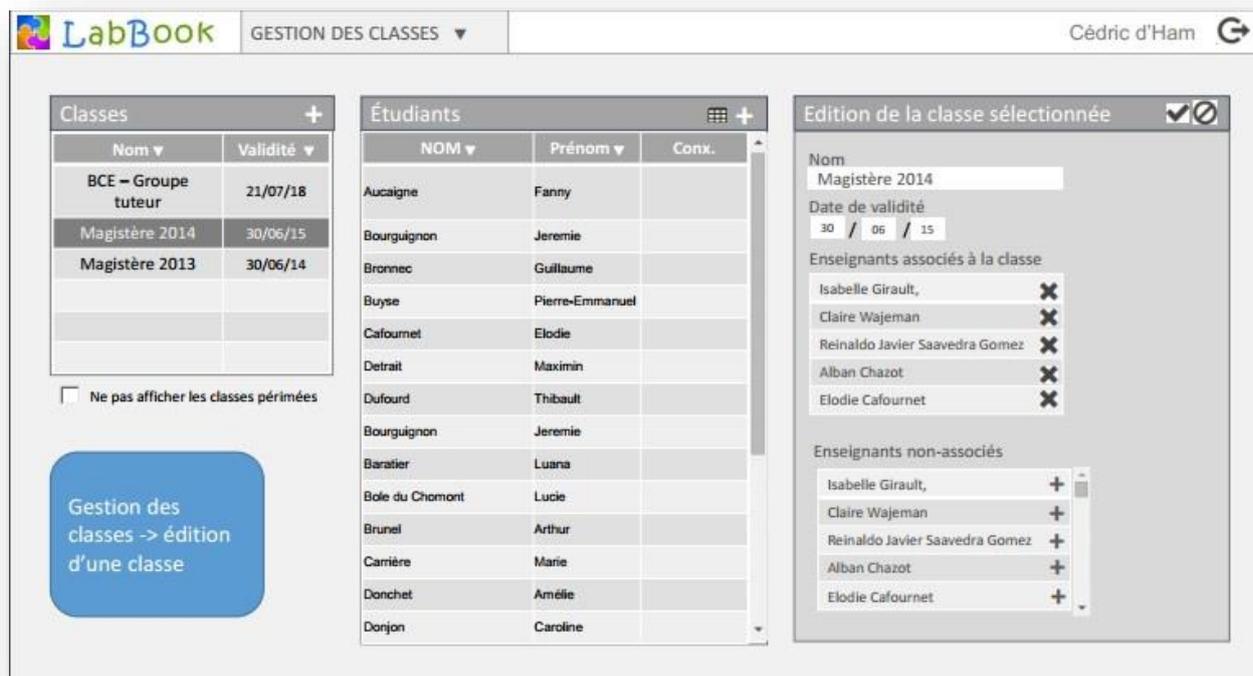


Figure 15 : Exemple de maquette de la page de gestion des classes.

Nous avons travaillé sur l'organisation des pages et leur facilité de prise en mains. Il fallait penser à la démarche d'un enseignant ; quelles sont les tâches sur lesquelles il va revenir plus régulièrement, quelles informations va-t-il vouloir et sous qu'elle forme ?

Avant de présenter chacune de ces pages dans les sections suivantes, je vais décrire les éléments qui ont été mis en place pour faciliter la prise en main pour les enseignants.

1. Le bandeau supérieur

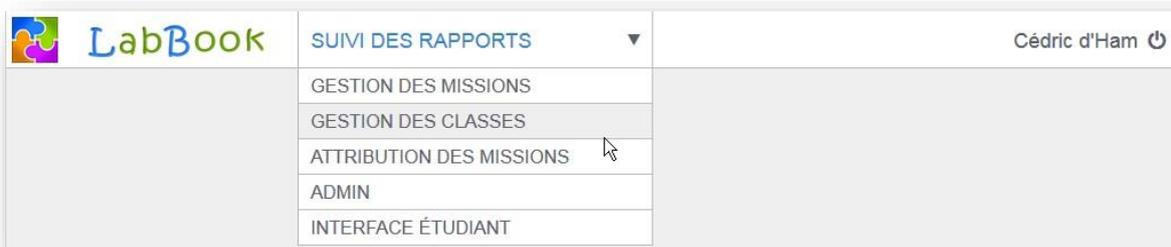


Figure 16 : Bandeau supérieur des pages dédiées aux enseignants (nouvelle interface).

J'ai proposé et mis en place un bandeau supérieur apportant plusieurs fonctionnalités. Un menu rappelle le nom de la page courante et permet d'accéder aux pages principales de l'enseignant. L'icône de l'application LabBook renvoie l'utilisateur sur la page de suivi des rapports que nous avons considéré comme étant la plus importante. Enfin, une icône à côté du nom de l'utilisateur propose la déconnexion et renvoie à la page d'authentification.

2. Une ergonomie générique pour les blocs et tableaux



Figure 17 : Ergonomie des entêtes de blocs et tableaux (nouvelle interface).

Les différents tableaux et blocs présentent une ergonomie unique (cf. Figure 17) :

- Un tableau principal avec une liste d'items et la possibilité d'ajouter un nouvel élément.
- Les éléments d'un tableau peuvent être triés dans un sens croissant ou décroissant par le clic répété sur l'entête de la colonne choisie.
- Certaines colonnes peuvent faire l'objet d'un filtre sur une valeur proposée dans une liste déroulante sous l'entête de la colonne.
- Un bloc optionnel secondaire d'informations décrit l'élément sélectionné du tableau principal. Des icônes proposent de modifier ou supprimer cet élément.
- Dans le cas d'une édition d'un élément créé ou sélectionné, un bloc d'édition permet de choisir et modifier les paramètres ou attributs de celui-ci. Ce bloc propose des icônes pour la validation ou l'annulation des modifications.
- Hormis le cas du bandeau supérieur, la couleur de fond blanche est réservée pour les éléments modifiables en édition.

Les réflexions sur l'ergonomie ont été liées à la logique et à la simplicité d'accès aux fonctionnalités. Ce travail intéressant m'a permis d'aborder des problématiques liées à la perception des différents utilisateurs face à un même élément. Cela a été en particulier le cas avec les icônes qui selon les systèmes d'exploitation des utilisateurs, et leur culture, peuvent être interprétés différemment (cf. Figure 18).

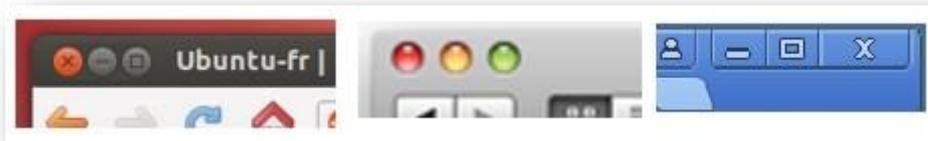


Figure 18 : Types d'icônes selon les systèmes d'exploitation, de gauche à droite : Linux, Apple et Microsoft.

Les applications web, du fait de leurs exécutions sur de multiples systèmes client, sont au cœur de cette problématique. Derrière ce thème, on retrouve la volonté de normaliser et de généraliser des ergonomies pour simplifier la prise en main de l'utilisateur [Google Developers15]. Ce mécanisme peut aussi avoir des effets négatifs par l'élimination de la variété au bénéfice de l'efficacité. Ce sujet, comme beaucoup d'autres abordés dans ce rapport, pourrait faire l'objet d'une étude approfondie au vu de son intérêt et sa complexité.

Dans le contexte de LabBook, qui est la recherche et l'enseignement, des références à l'ergonomie d'Apple peuvent être trouvées. C'est le cas des icônes de fermeture des fenêtres de ressources et commentaires qui représentent un trait bas.

Le travail dans LabBook est donc d'utiliser le même référentiel d'icônes dans toute l'application pour en garder la cohérence.

B. Gestion des classes et de leurs étudiants.

Un objectif des pages de gestion pour les enseignants, était de maintenir l'ensemble des fonctionnalités sur un seul écran. Cette problématique s'est présentée dans la page de gestion des classes. Il fallait gérer les classes et les étudiants de ces classes avec toutes les options d'affichage, d'édition et de validation. J'ai donc placé deux blocs fixes sur la gauche et un bloc contextuel sur la droite. Cela permet d'afficher dans ce bloc contextuel les informations nécessaires, selon l'élément sélectionné, la classe ou l'étudiant d'une classe préalablement choisie.

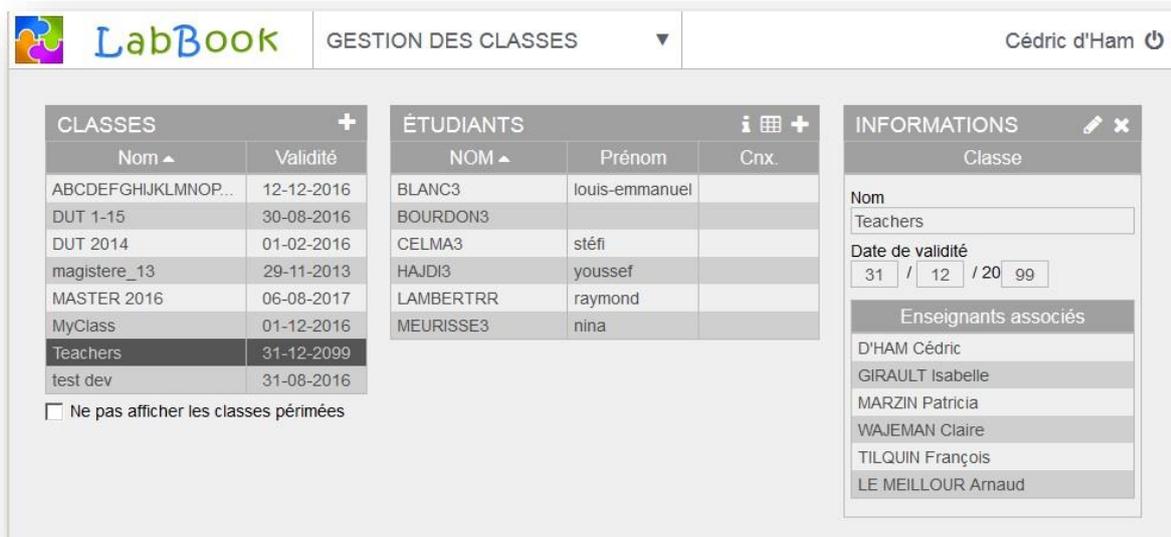


Figure 19 : Page de gestion des classes (nouvelle interface).

L'ancienne interface souffrait de redondance dans l'affichage des icônes. Dans cette nouvelle page, ce problème n'existe plus.

La page permet un accès rapide à toutes les fonctions, de l'ajout d'une classe ou d'un étudiant, à l'affichage de leurs informations et la modification de celles-ci.

J'ai aussi implémenté l'importation des étudiants depuis un fichier. Afin de sécuriser l'enregistrement dans la base, j'ai ajouté une étape de validation des données importées du fichier (cf. Figure 20). J'ai aussi autorisé la modification directe de celles-ci dans les champs afin d'apporter plus de souplesse aux enseignants.

Dans cette interface, un enseignant peut ajouter un ou plusieurs nouveaux étudiants, mais il ne peut pas en ajouter depuis d'autres classes. Cette fonctionnalité n'était pas souhaitée car jugée non nécessaire dans le contexte actuel d'utilisation. Au besoin, une évolution pourra proposer une importation depuis les étudiants déjà enregistrés. Il ne faudra alors, que reprendre la fenêtre d'importation existante et y ajouter les options de filtre et de tri telles qu'implémentées dans les autres pages.

IMPORTATION							
Sélectionnez les lignes à importer à l'aide des cases à cocher. Vous pouvez modifier le contenu des champs avant de valider. *Champs obligatoires : Nom et/ou Prénom, Login, Mot de passe.							
<input type="checkbox"/>	NOM*	Prénom*	Date de naissance	E-mail	Organisation	Login*	Mot de passe*
<input type="checkbox"/>	Romain	jean	12/01/1981	thomas@free.fr		toto05	toto05
<input type="checkbox"/>	tarot	francis	03-21-1986	roberdu-srd@free.fr	ENS3	toto04	toto04
<input checked="" type="checkbox"/>	lambert	raymond	04/06/1984	rlambert@gmail.com	DUT	toto06	toto06
<input checked="" type="checkbox"/>	bourdon		12/01/1981		DUT	toto08	toto08
<input checked="" type="checkbox"/>	Martin-laval	Pierre-Francois	04/06/1984	roberdu-srd@free.fr		toto09	toto09
<input checked="" type="checkbox"/>	ducret	Arnaud	12/04/1982	aducuret@free.fr		toto10	
<input type="checkbox"/>	blanc	Louis-emmanuel	12/01/1981	roberdu-srd@free.fr	BTS	toto13	toto13
<input type="checkbox"/>	Meurisse	nina	12/04/1982		BTS	toto14	toto14
<input checked="" type="checkbox"/>	salvador	thomas	04/06/1984	roberdu-srd@free.fr	BTS	toto15	toto15
<input type="checkbox"/>	hayek	salma		shayek@gmail.com		toto17	toto17
<input type="checkbox"/>	henderson	shirley	12/04/1982				toto18
<input type="checkbox"/>	bentivoglio	fabrizio	12/01/1981			toto19	toto19
<input type="checkbox"/>	girardello	ludovico	12/01/1981			toto21	toto21

Figure 20 : Fenêtre d'importation d'étudiant par fichier au format csv (nouvelle interface).

C. Gestion des groupes

L'ancienne interface fusionnait la gestion des groupes et leur suivi. De plus, il fallait passer par la mission pour accéder au groupe, il n'y avait donc pas de vue synthétique de ceux-ci. Avec mon responsable, nous avons décidé de séparer le suivi des rapports des groupes, de leur gestion. Des groupes sont créés à des moments spécifiques, comme les classes, mais le suivi des rapports est l'activité récurrente de l'enseignant dans son interface générale.

Cette page pouvait s'appeler « gestion des groupes », « création des rapports » ou « attribution des missions » car ces trois actions sont effectuées ici.

Nous avons choisi de nommer cette page « *Attribution des missions* » pour reprendre la logique de l'interface des étudiants pour les quels des missions sont affectées.

La page d'attribution des missions permet donc de visualiser l'ensemble des groupes (cf. Figure 21). L'enseignant peut ici créer, modifier et supprimer un groupe et le rapport lié.

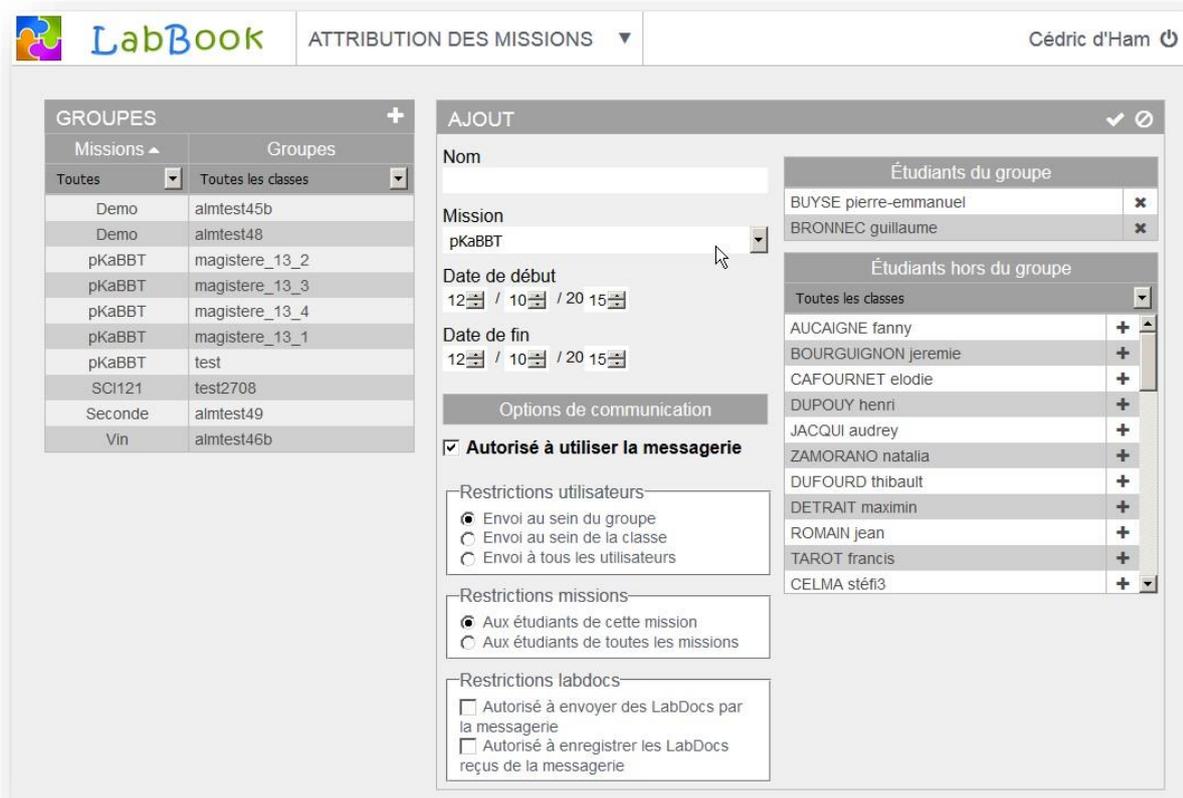


Figure 21 : Page d'attribution des missions (nouvelle interface).

L'ajout de membres d'une liste à une autre liste utilise la même ergonomie sur les différentes pages. Les listes sont superposées avec la liste cible en position supérieure. Une icône sur la droite de chaque élément propose et spécifie par sa forme l'action de l'ajouter ou le retirer de la liste cible. Dans cette page, ce type de liste permet l'ajout ou le retrait au groupe édité des élèves inclus dans les classes de l'enseignant connecté.

D. Gestion des missions.

La page de *gestion des missions* (cf. Figure 22) correspond à la page de *choix des missions* initiale. On y retrouve avec la nouvelle ergonomie ; les informations de chaque mission sur son statut, sa description et les enseignants associés. Les icônes proposent de créer, tester ou dupliquer une mission, et si le statut de l'enseignant est *concepteur* s'y ajoutent les options de modification et suppression.

Cette page permet d'accéder à la page d'édition des missions lorsque l'on souhaite modifier, dupliquer ou créer une mission. Dans le cas de la création, on doit en premier lieu, valider dans cette page, l'enregistrement du code et du nom de la nouvelle mission.

The screenshot shows the 'LabBook' interface for 'GESTION DES MISSIONS'. The user is identified as 'Cédric d'Ham'. The main table lists various missions with columns for 'Rôle', 'Code', 'Nom', 'Statut', and 'Création'. The 'Xeroderm' mission is highlighted. To the right, the 'INFORMATIONS' panel provides details for the selected mission, including its code, name, status, creation date, and a detailed description of the project.

Rôle	Code	Nom	Statut	Création
Tous			Toutes	
Concepteur	DivGen	Diversité génétique et survie	Publique	
Concepteur	SCI121	Dosage du fer total dans des eaux ferrugineuses	Publique	09-07-2010
Concepteur	Cin	Cinétique	Publique	12-07-2010
Concepteur	Xeroderm	Etude du Xeroderma pigmentosum (mutations du gène Xpa)	Publique	13-03-2013
Concepteur	Demo	Comment fait-on le thé en France et au Japon ?	Publique	18-01-2011
Concepteur	Vin	Dosage de substances dans le vin	Publique	29-02-2012
Concepteur	Clepsy	Clepsydras	Publique	24-01-2011
Concepteur	Dipole	Dipôles électriques	Publique	14-03-2011
Concepteur	Seconde	Modification de l'information génétique	Publique	02-11-2010
Tuteur	pKaBBT	Diagramme de prédominance d'un indicateur coloré	Publique	03-01-2011
Tuteur	Biodiv	Biodiversité d'une pelouse	Publique	29-02-2012
Observateur	Ac-Ag	Liaison anticorps-antigène et structures tridimensionnelles	Publique	15-03-2011
Observateur	Eaux-Fe	Eaux ferrugineuses	Publique	29-02-2012

INFORMATIONS

Code **Xeroderm**

Nom **Etude du Xeroderma pigmentosum (mutations du gène Xpa)**

Statut **Publique**

Date de Création **13-03-2013**

Dernière modification **13-03-2013**

Description

Le *Xeroderma pigmentosum* est une maladie génétique qui s'exprime lorsque certaines protéines responsables du système de réparation de l'ADN sont défectueuses (protéine Xpa par exemple). Cette mission concerne l'étude de 6 allèles codant pour la protéine Xpa. En suivant une démarche expérimentale, vous devez résoudre le problème suivant :

Comment expliquer que des mutations du gène Xpa (allèles Xpa_1 à Xpa_7) aient des conséquences sur la taille et la fonctionnalité des différentes protéines synthétisées ?

Sur la base de l'analyse des ressources présentées vous devez réaliser le travail suivant :

Formuler des hypothèses concernant l'origine des différents types de protéines

Concevoir des protocoles afin de tester chaque hypothèse formulée

Évaluer les protocoles conçus

Enseignants associés

D'HAM Cédric	Concepteur
HÉRITIER Claudine	Tuteur
TILQUIN François	Tuteur

Figure 22 : Page de gestion des missions (nouvelle interface).

E. Suivis des rapports

Un enseignant peut être amené à passer du temps lors de la création de ses missions, mais l'activité récurrente qu'il va mener est de suivre le travail de ses groupes d'élèves.

La page de suivi de rapports remplit cette fonction (cf. Figure 23). À la différence de l'ancienne interface, tous les groupes sont à présent visibles. Les options d'affichage par tri et le filtre sur une mission ou une classe spécifique facilitent l'accès aux informations.

Les fonctionnalisées de suivis sont les mêmes que dans la précédente interface de gestion des groupes :

- Possibilité d'écrire un email aux membres du groupe par un lien de type *MailTo*.
- Modification du statut du rapport vers les états *en cours*, *rendu* et *archivé*.
- Accès aux rapports qui ont été modifiés, dans un mode de *lecture*. L'enseignant peut néanmoins laisser des commentaires sur les LabDocs.
- Accès au service de messagerie de LabBook.

RAPPORTS		Début	Fin	Avancement	LD	Dernière modif.	
Missions ▾	Groupes						
Toutes ▾	Toutes les classes ▾			Tous ▾			
Cin	Grp_132 (n. zamorano)	16-08-2015	16-10-2017	Non débuté ▾	0		0
Demo	Grp_XX1 (c. d'ham, a. jacqui, m. detrait) ✉	16-08-2015	16-10-2016	Rendu ▾	0		0
Demo	almtest45b (p. e. buyse, e. cafournet, a. jacqui, n. zamorano)	31-07-2015	02-03-2016	Non débuté ▾	0		0
Demo	almtest48 (t. dufourd, j. romain, r. lamberti) ✉	31-07-2015		Non débuté ▾	0		0
pKaBBT	magistere_13_4 (p. e. buyse, n. zamorano, t. dufourd)	04-02-2013	04-03-2013	Rendu ▾	0		0
pKaBBT	magistere_13_2 (f. aucaigne, h. dupouy)	04-02-2013	04-03-2013	Archivé ▾	0		0
pKaBBT	magistere_13_3 (j. bourguignon, a. jacqui)	04-02-2013	04-03-2013	Archivé ▾	0		0
pKaBBT	magistere_13_1 (g. bronnec, e. cafournet)	04-02-2013	04-03-2013	En cours ▾	0		0
pKaBBT	test (m. detrait)	01-06-2015		Non débuté ▾	0		0
SCI121	Grp_133 (a. jacqui)	16-04-2015	16-10-2017	Non débuté ▾	0		0
SCI121	test2708 (j. bourguignon, a. follet) ✉	27-08-2015	01-10-2015	En cours ▾	1	10-09-2015 - 18:03	0
Seconde	almtest49 (h. dupouy, s. celma, r. lamberti)		01-01-2015	Non débuté ▾	0		0
Vin	almtest46b (f. aucaigne, g. bronnec, e. cafournet, h. dupouy)	01-06-2015	09-09-2015	Non débuté ▾	0		0

Avancement du rapport
 Non débuté : la mission n'a pas été consultée.
 En cours : les étudiants ont accédé au moins une fois au rapport.
 Rendu : les étudiants attendent une validation de leur rapport par l'enseignant. Ils ne peuvent plus le modifier.
 Archivé : le rapport est terminé et non modifiable.

Figure 23 : Page de suivi des rapports (nouvelle interface).

F. Ajout de la fonctionnalité d'édition des missions

Cette page est totalement différente des autres pages spécifiques aux enseignants, qui sont dédiées à des opérations de gestion. Elle constitue le cœur de l'application, avec la page d'édition des rapports de l'étudiant.

Cela aurait pu être un formulaire de saisie *standard*, mais je souhaitais produire une interface agréable et ergonomique. Je me suis inspiré d'une interface provenant du site de Google contact. J'ai produit les premières maquettes sans avoir idée des techniques que j'allais utiliser pour les mettre en œuvre.

La plupart des éléments sont enregistrés *à la volée*, soit directement à la sortie du champ mis à jour, soit par des validations d'enregistrements par éléments. L'enseignant n'a pas une option d'enregistrement global de la page tel un formulaire standard. Ceci a été décidé pour lui laisser une plus grande souplesse d'utilisation et diminuer les risques de perte de données.

1. Paramètres de la mission

La première partie de la page concerne les paramètres de la mission (cf. Figure 24). On y retrouve le code, le nom et la description de la mission. J'ai réutilisé l'outil de traitement de texte des LabDocs pour le travail sur les textes mis en forme dans cette page (texte de présentation et consigne courte). C'est dans cette partie que l'on gère l'association des enseignants à la mission et le statut de la mission. Une mission peut avoir trois statuts :

- **Création**, la mission peut être modifiée librement, mais ne peut pas être attribuée à des groupes d'étudiants.

- **Privé ou public**, la mission peut être attribuée à des groupes d'étudiants ; seuls certains aspects de la mission peuvent être modifiés (tout ce qui n'ampute pas les rapports des groupes). Une fois que la mission a été attribuée à un groupe, son statut ne peut plus revenir en "Création".
- **Privé**, la mission ne peut être vue que par les concepteurs et les tuteurs de la mission.
- **Public**, la mission peut être vue par tous les enseignants connectés à LabBook.

The screenshot shows the 'EDITION DE MISSION' page in LabBook. At the top, there is a header with the LabBook logo, the title 'EDITION DE MISSION', a user profile for 'Cédric d'Ham', and an eye icon. Below the header, the section 'PARAMETRES DE LA MISSION' is displayed. It contains several input fields: 'Code' (Xeroderm), 'Nom' (Etude du Xeroderma pigmentosum (mutations du gène Xpa)), and 'Statut' (Public). There is also a section for 'Enseignant(s) associé(s)' with a table listing 'Claudine Héritier' and 'François Tilquin' as tutors. A 'Texte de présentation' section contains a detailed description of the Xeroderma pigmentosum mission, including a list of tasks for students.

Figure 24 : Première partie de la page d'édition des missions, dédiée aux paramètres de la mission.

2. Les ressources

La partie suivante permet de gérer les ressources que l'enseignant va joindre à la mission. Outre l'ajout d'un texte formaté, cette section de la page permet de télécharger des fichiers sur le serveur.

Il y avait ici des listes d'items pour lesquels plusieurs actions étaient liées. Nous devons éviter la répétition d'icônes. J'ai considéré deux méthodes ; soit une série d'icônes apparaissaient contextuellement au passage sur l'item soit un icône unique était répété pour chaque ligne, donnant accès à un menu contextuel selon l'item. Cette dernière solution a été choisie et utilisée à différents endroits de la page (cf. Figure 25).

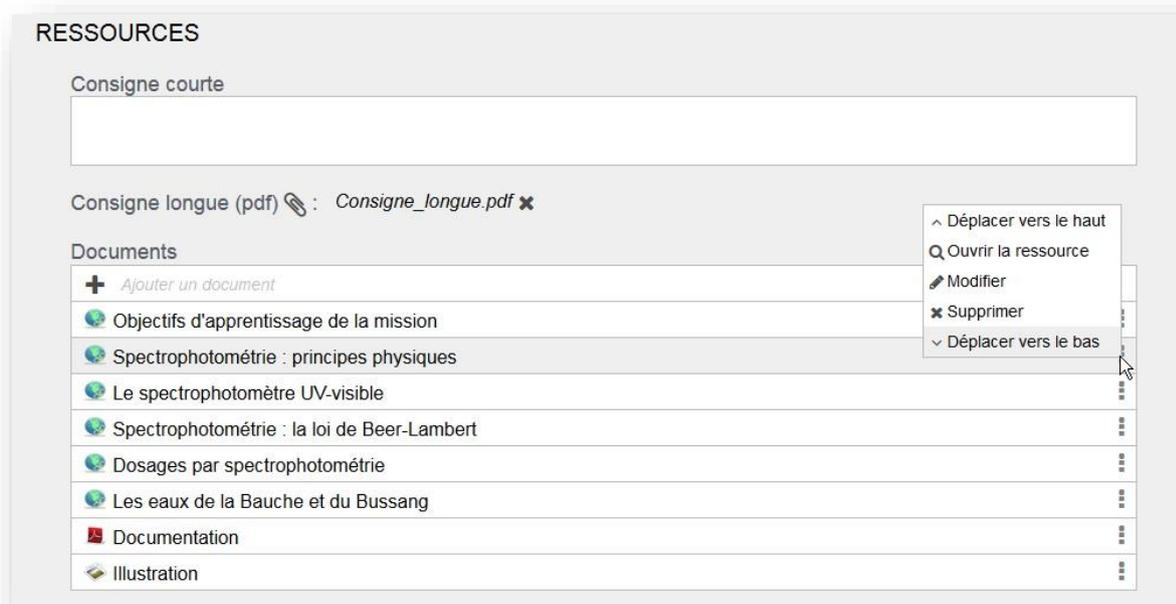


Figure 25 : Deuxième partie de la fenêtre d'édition des missions, dédiée aux ressources.

Les documents de ressources sont soit des liens vers des pages internet, soit des fichiers de type *image* ou *pdf*. Cette section permet d'effectuer les opérations habituelles (ajout, consultation, modification et suppression) et de gérer l'ordonnancement.

3. Le rapport

La section suivante est la principale car elle gère la structure du rapport (cf. Figure 26).

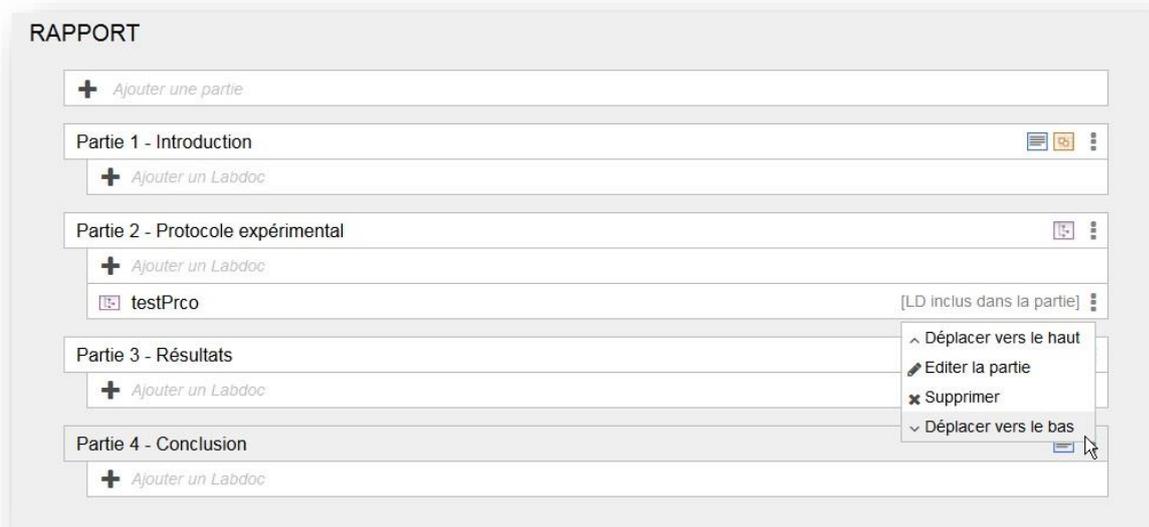


Figure 26 : Troisième partie de la page d'édition des missions, dédiée au rapport.

L'ergonomie de présentation des items multiples (enseignants associés, document de ressources, parties de rapport et LabDocs) est identique. On retrouve un bloc vide contenant une icône et un commentaire proposant l'ajout d'un item. Les détails des items ne sont pas affichés, c'est lors de leur édition que des formulaires apparaissent à la place des items et proposent toutes les options.

L'édition d'une partie permet de gérer son titre, la consigne donnée à l'étudiant et les types de LabDocs que les étudiants peuvent créer dans cette partie (cf. Figure 27).

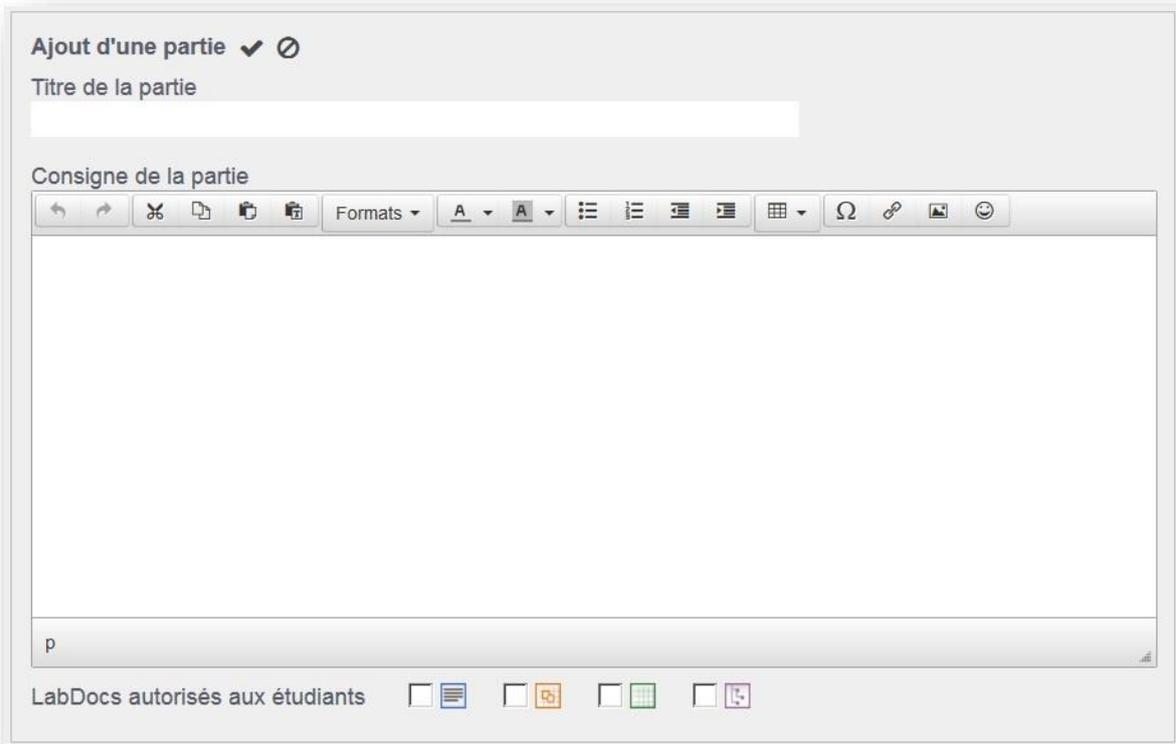


Figure 27 : Ajout d'une partie de rapport dans la fenêtre d'édition des missions.

La création d'un LabDoc mis en place par l'enseignant s'effectue en deux temps :

- Création du LabDoc en précisant son nom et le type. Le type du LabDoc est un choix libre pour l'enseignant, indépendamment de ceux autorisés à l'étudiant dans la partie. La place d'un LabDoc créé par l'enseignant peut être soit dans la partie de rapport sélectionnée, soit dans le menu. L'étudiant qui crée un LabDoc dans son rapport, a un menu lui proposant les différents types de LabDoc *vides* qui lui sont autorisés. Les LabDocs de menu préétablis doivent apparaître dans ce menu de création et respecter le type autorisé à l'étudiant. L'interface étudiant n'a pas encore cette nouvelle fonctionnalité implémentée, les options ont donc été temporairement cachées dans l'interface enseignante.

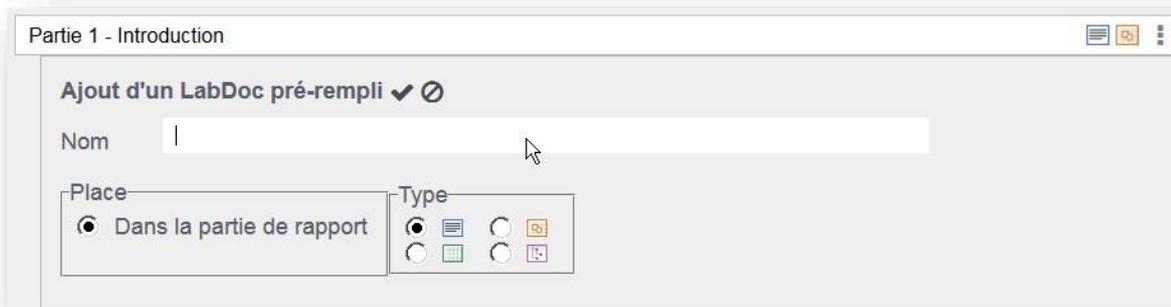


Figure 28 : Ajout d'un LabDoc de l'enseignant dans la fenêtre d'édition des missions.

- Edition du LabDoc ou l'utilisateur peut modifier le nom et préciser les *autorisations* qu'aura l'étudiant sur le LabDoc dans son rapport (cf. Figure 29).

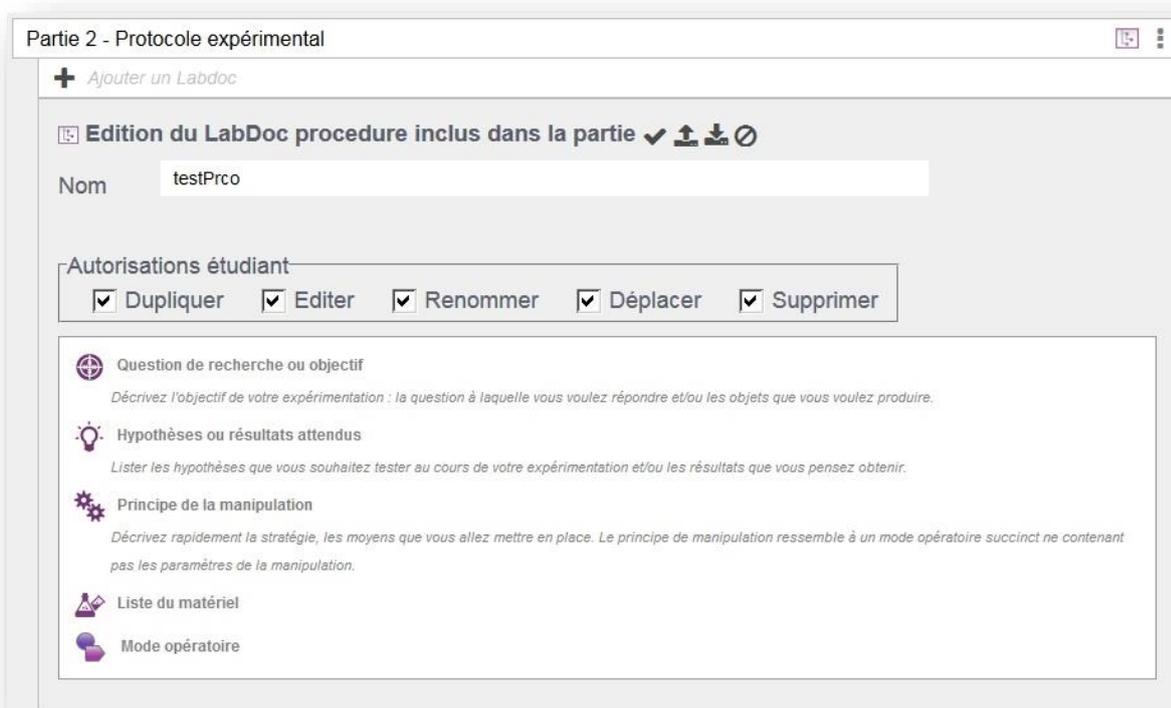


Figure 29 : Édition d'un LabDoc de l'enseignant dans la fenêtre d'édition des missions.

Dans ce formulaire, l'outil de création de LabDoc est chargé et permet à l'enseignant d'y effectuer toutes les modifications souhaitées.

Le LabDoc de création de protocole expérimental est particulier car construit au format XML, il autorise l'enseignant à des modifications importantes. Ces modifications ne sont pas gérées par l'outil de protocole qui est complètement dédié à l'étudiant utilisateur. L'enseignant doit travailler pour l'instant directement sur le code XML avec un éditeur adéquat. Ceci nécessite que l'enseignant puisse télécharger le contenu du LabDoc, et le renvoyer vers le serveur. Deux icônes ont été ajoutés à cette fin dans le formulaire d'édition.

IV. Conclusion de la première partie

La compréhension de l'application s'est faite tout au long du stage. Les concepts standards, telle la gestion d'une classe, s'apprennent facilement, mais les processus métiers ont requis plus d'explications. Cette compréhension était nécessaire pour que l'évolution s'intègre dans l'existant.

L'attention s'est portée particulièrement sur l'ergonomie et il était nécessaire d'allier des objectifs distincts :

- Garder une interface claire et intuitive.
- Offrir les fonctionnalités permettant à l'utilisateur de faciliter sa tâche et d'atteindre ses propres objectifs.
- Offrir une application réactive.

Dans cette démarche, le travail sur les maquettes a été déterminant pour concevoir des pages simples et fonctionnelles. J'ai dû adapter mes idées avec les contraintes imposées, faire preuve de créativité et de sens pratique tout en restant ouvert aux habitudes et besoins de l'utilisateur. Le succès d'une telle évolution est donc aussi le fruit d'une bonne communication et du travail sur la qualité des échanges.

Lors de la création de la première maquette, je n'avais aucune idée des moyens et techniques pour la réaliser. La démarche de conception des interfaces est différente lorsque l'on inclut consciemment ou pas les problématiques de codage.

L'essentiel des interfaces créées est le fruit de mes propositions, ce qui est une grande satisfaction. Je n'ai pas seulement mis en œuvre des évolutions mais j'ai pu aussi contribuer à leurs élaborations.

DEUXIEME PARTIE - PROBLEMATIQUES TECHNIQUES

L'activité au cœur d'un stage de développement d'application est la production de code. Un mémoire d'ingénieur n'a pas pour but de revenir sur les détails du code produit comme dans une documentation technique. Le fait est que cette composante est loin d'être triviale et certaines de ces problématiques méritent d'être soulevées.

Je présenterai tout d'abord, l'architecture logicielle générale de LabBook, pour ensuite expliquer le travail effectué sur la gestion des données. Le chapitre suivant sera consacré aux Cascading Style Sheet (CSS) et conventions de nommage ainsi qu'aux enjeux afférant aux stratégies choisies. Je conclurai cette partie en présentant le travail effectué sur les pages enseignantes.

I. Architecture de LabBook

L'application a été développée par étapes et n'utilise pas de cadre structurant comme Angular JS, Zend, Symfony ou Laravel.

La structure principale de fichiers du projet LabBook est la suivante :

- **authoring_tools** : Contient les pages dédiées aux interfaces des enseignants.
- **common** : Contient des fichiers PHP (*Hypertext Preprocessor*) intégrant des paramètres et des fonctions générales de contrôle d'accès et d'écriture sur la base de données.
- **functions_js** : Contient les fonctions JavaScript mutualisées des pages principales.
- **functions_php** : Contient les fonctions PHP mutualisées des pages principales.
- **libraries** : Ce dossier contient les bibliothèques d'utilitaires comme jQuery qui sont utilisés dans les pages de LabBook, ainsi que le plugin TinyMCE qui est l'un des outils de création de LabDoc.
- **pages** : Sont placées ici les pages principales du site avec leurs images et fichiers en charge du style.
- **tools utilities** : Contient les fonctions (JavaScript et PHP) communes aux différents outils modulaires tels que COPEX (protocole) ou ZWIBBLER (dessin).
- **learner_docs** : Sont placés ici, les documents de ressources ajoutés par les étudiants.
- **missions** : Contient les fichiers (de type *pdf* ou *image*) de ressources et les consignes longues téléchargées par le concepteur de la mission.

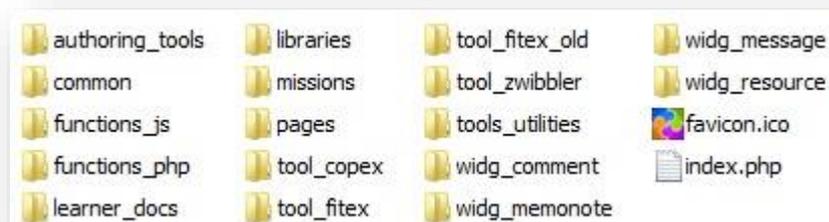


Figure 30: Dossiers à la racine de l'application LabBook

Les autres dossiers concernent les outils et les modules avec pour chacun leur propre organisation de fichiers.

A. Articulation initiale des pages

Dans son état initial, l'interface enseignante avait pour base celle des étudiants à laquelle s'ajoutaient des fonctionnalités spécifiques comme la gestion des classes et des groupes.

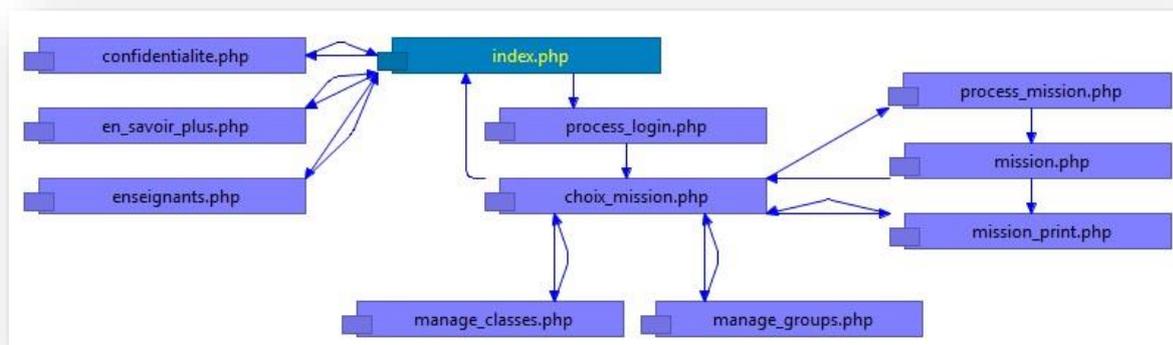


Figure 31: Structure initiale des pages de l'application LabBook

Si l'on observe la structure des pages (cf. Annexe F-2-3-4-5), nous pouvons constater que les méthodes de constructions sont différentes. Selon les pages, le code JavaScript est soit intégré directement dans la page PHP soit externalisé dans un fichier propre d'extension « *.js* »

De même pour le code PHP, selon les pages, les fonctions effectuant des traitements sont gérées de manière différente. Elles peuvent être rassemblées dans des pages dont le nom se termine par « *_fn* », d'autres fois distribuées dans des fichiers dédiés au **Vues** ou au **Modèle**. Dans le cas de la page « *mission.php* », des requêtes SQL et traitements sont directement intégrés à celle-ci.

Les requêtes AJAX provenant du code JavaScript sont envoyées vers des fichiers se terminant par « *_ajax.php* » ou « *_fn.php* ».

Tous ces éléments sont la trace des évolutions de l'application pour laquelle cette architecture s'est adaptée aux situations. Il en ressort une structure où les pages principales sont regroupées par type et les pages dédiées aux modules rassemblées par module.

B. Méthode inspirée de MVC

Le modèle MVC est une méthode d'organisation qui a pour objectif de distinguer les parties du code selon leurs objectifs.

- **Modèle** : cette partie gère les *données* du site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le Contrôleur. On y trouve donc les requêtes SQL. Le Modèle informe la Vue de changement et répond aux requêtes.
- **Vue** : cette partie se concentre sur l'*affichage*. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions très simples. Elle envoie au Contrôleur les événements de l'utilisateur.
- **Contrôleur** : cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le Modèle et la Vue : le Contrôleur va demander au Modèle les

données, les analyser, prendre des décisions et renvoyer le texte à afficher à la Vue. Il demande aussi les mises à jour vers le Modèle.

Ce type d'organisation cloisonne les parties de l'application, ce qui permet d'effectuer plus facilement des modifications. Cette indépendance entre les parties permet aussi une grande souplesse et facilite le travail en parallèle de différentes équipes de développement. La séparation par type de données clarifie l'architecture et permet d'affiner les règles de sécurité pour l'accès aux fichiers. Tous ces éléments font du patron de conception MVC, une bonne pratique de développement.

Sans l'aide d'un cadriciel, la mise en place et le maintien d'un modèle MVC est un exercice supplémentaire et une contrainte importante. L'implémentation de ce modèle tel que j'ai pu le voir dans différents tutoriaux se fait dans le cadre d'un développement en mode objet et augmente considérablement le nombre de fichiers. L'utilisation de différentes techniques accompagnent les exemples mis en ligne, comme la réécriture d'URL.

Une contrainte du projet était de ne pas utiliser de développement en mode objet, et de rester sur un style procédural.

LabBook est une application créée en 2010, qui a donc une histoire et dont l'architecture dépend des choix de conception initiaux. Mon intervention devait se concentrer sur une partie de cet applicatif et n'impliquait pas de réingénierie. Dans ces conditions, il ne m'était pas possible de justifier une prise de risque importante accompagnée d'un temps de développement conséquent. J'ai tout de même pu agir sur les développements qui m'incombaient, c'est pourquoi j'ai saisi l'opportunité de remplacer le mode de fonctionnement historique pour ne faire transiter que les données et non des morceaux de vue.

Contrairement à l'approche « table rase » qui consiste à tout reconcevoir, cette approche d'amélioration continue, plus douce, pose les premières bases d'un mécanisme architecturé et sain. Bien entendu, il est de fait que la structuration d'une application existante est un exercice difficile car elle implique un gros investissement sans gain pour l'utilisateur.

C'est pour ces raisons qu'il n'est pas implémenté de manière « classique » sur les pages et qu'il a été adapté au contexte, en autorisant par exemple, les requêtes directes en base de données dans les fichiers de Vue.

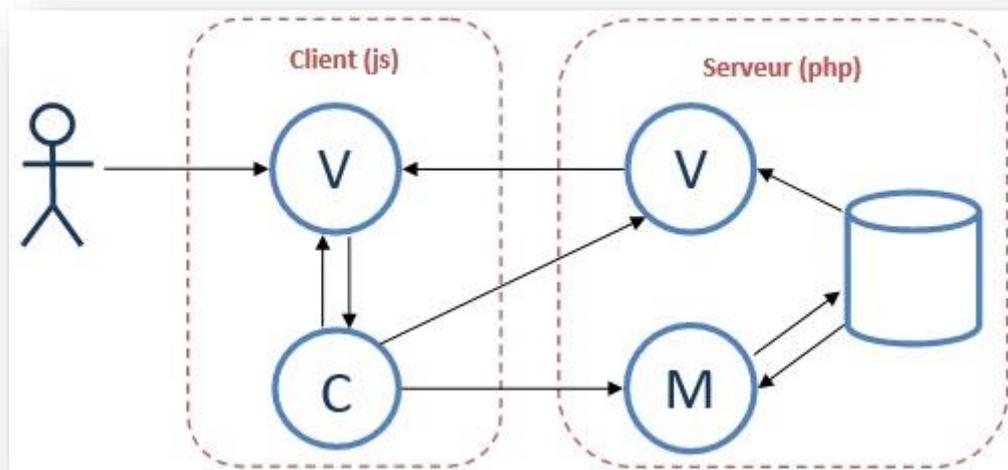


Figure 32: Implémentation du modèle MVC dans LabBook

Les outils Fitex et Copex comportent des fichiers JavaScript dédiés aux Modèles car ils manipulent des données locales à la page. Nous ne sommes pas exactement ici dans la situation où l'on accède à la base de données MySQL.

Au final, l'objectif dans LabBook a été d'isoler le code accédant aux données (Modèle), celui comportant de l'HTML dédié à la Vue, et les traitements associés au Contrôle, que ce soit au sein de fichiers ou de fonctions spécifiques. Cette approche nous a permis de gagner en clarté et de faciliter les modifications. La gestion de l'affichage et sa maintenance requièrent des modifications dans le code HTML. Si celui-ci est éparpillé et redondant, il est complexe de le comprendre et les risques d'erreurs augmentent.

Cette organisation de l'existant permet de faciliter la maîtrise du code en attendant que les améliorations successives fassent disparaître peu à peu les morceaux de vue de la partie PHP. La mise en place d'un modèle MVC sur l'ensemble devra vraisemblablement faire l'objet d'un travail de fond, itératif et incrémental. Cela limite l'investissement et les risques, tout en augmentant chaque jour la qualité.

C. Gestion des composants modulaires

Plusieurs composants sont greffés à la page « *mission.php* », tels que les outils de création de LabDocs (Copex, Fitex, Drawex, TinyMCE) et les services de commentaires, de messagerie ou de ressources. Leur fonctionnement est atypique car ils pourraient être chargés depuis d'autres pages, ce qui a été le cas avec les outils de LabDoc dans la page d'édition des missions, et le service de messagerie dans la page de suivi des rapports.

Cette intégration dans de nouvelles pages pose certains problèmes dus à la modification du contexte. Un composant modulaire de ce type peut avoir des problèmes liés à l'espace de nom, ce type d'anomalie correspond à des conflits entre éléments (fonctions, classes, identifiant) portant le même nom. Comme soulevé dans le chapitre sur les conventions de nommage, des *fonctions* ou *classes* portant les mêmes noms peuvent se parasiter et provoquer des événements inattendus.

Globalement, les services de messagerie et de commentaires sont indépendants, car ils ne nécessitent pas d'adapter la page hôte. Ils gèrent leurs événements et leurs accès aux données. La

seule précaution à prendre étant d'être vigilant sur la propagation de règles de style pouvant atteindre de manière non souhaitée ces composants. En utilisant des préfixes pour les classes et identifiants HTML, les risques de conflits ont été réduits.

Les outils de création de LabDocs ont, quant à eux, été spécialement mis en place pour la page d'édition des rapports. Des fonctions ayant pour tâche de gérer leurs comportements, ont été mutualisées dans des fichiers centralisés. Les données de différents outils sont stockées dans un tableau de la page hôte. Toutes ces dépendances complexifient l'intégration dans d'autres pages, comme cela a été le cas lors de l'élaboration de la page d'édition des missions.

Il existe des patrons de conception [OSMANI12] proposant des solutions pour faciliter l'organisation modulaire d'une application. Ce genre d'approche demanderait un investissement important en réingénierie.

D. Conclusion

Ce chapitre a présenté l'application et les méthodes choisies au long des dernières années pour la faire évoluer. Malgré les exigences de simplicité requises pour sa maintenance, la succession des interventions effectuées par des personnes aux profils différents, n'a pas permis d'obtenir une logique complètement homogène dans son architecture.

A défaut d'effectuer une refonte de l'application, la démarche des interventions sur celle-ci est alors de ne pas alourdir cette hétérogénéité par l'apport de techniques complexes, aussi bonnes soient-elles. Comme l'expriment les exigences de conceptions formulées par mon responsable, je devais m'inspirer de l'existant et utiliser des techniques simples.

Le développement d'une application comme LabBook s'effectue sur plusieurs années. Depuis sa création, cinq années se sont écoulées. Les technologies web évoluent rapidement, il faut donc arbitrer entre l'utilisation de technologies et outils qui peuvent apporter un gain de temps ou aider à structurer l'application, et rester sur des développements moins complexes mais mieux maîtrisés.

II. La base de données

L'annexe V.E.1 présente l'organisation des tables de la base de données. C'est un des premiers documents qui m'a été fourni, pour comprendre le fonctionnement de l'application. L'exigence la concernant était d'éviter d'intervenir sur la structure, car une fois qu'un élément est en place, toute modification peut demander un gros travail de réingénierie.

Les relations entre les tables (cf. Figure 33) illustrent l'organisation des utilisateurs en classes, et le découpage des missions et rapports (*mission.conf*). La structure de la base de données porte les informations essentielles de l'application.

Deuxième partie – Problématiques techniques

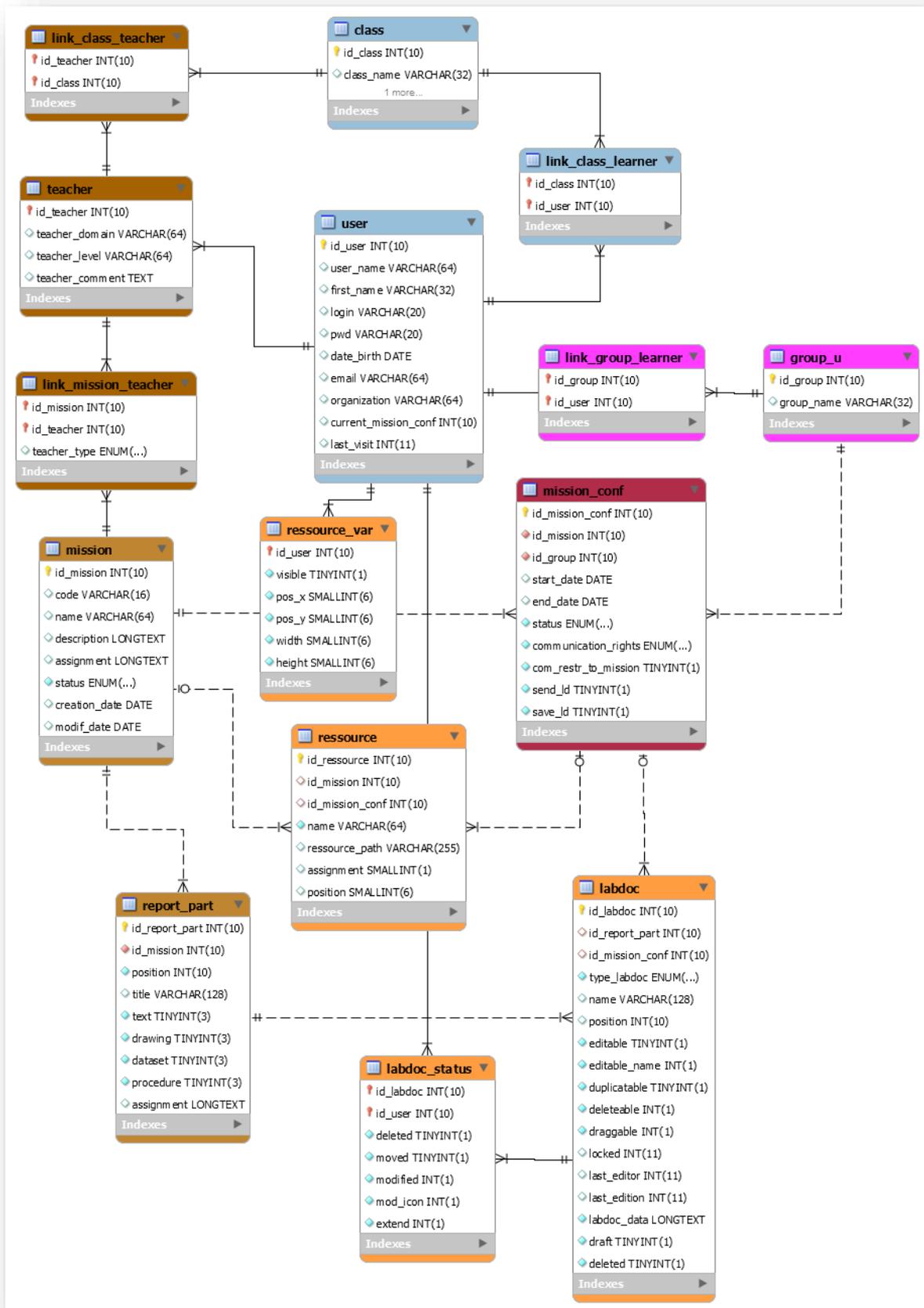


Figure 33 : Diagramme EER de la base de données LabBook, hors tables liées à la messagerie et aux commentaires

A. Mise à jour des méthodes d'accès

Très rapidement, j'ai averti mon responsable que l'extension de PHP, MySQL et ses fonctions étaient obsolètes et allaient être supprimées dans la prochaine version de PHP. Toutes les fonctions actuelles de l'application s'appuient sur cette extension.

Dans l'optique de rester au plus proche du code actuel, il a été décidé que je redéveloppe les fonctions d'accès à la base de données en utilisant l'extension MySQLi dans sa forme procédurale. Par conséquent, j'ai été aussi indépendant du reste de l'application pour ce type d'accès. J'ai créé pour cela le fichier « *auth_cnx_db.php* » et ses fonctions.

Tableau 1 : Liste des fonctions d'accès à la base de données

dbConnect()	dbPing()	dbClose()
dbQuery()	dbRealEscapeString()	dbInsertId()
dbUnconnect()	dbFetchAssocAll()	dbClean()

Une autre option aurait été d'utiliser PDO (PHP Data Object) [ROSSOLINI08], qui est une autre extension de PHP pour la gestion des bases de données. Celle-ci apporte plusieurs avantages que sont l'abstraction au système de gestion de la base utilisée, et un système de déclaration préparée. Une instruction préparée est une déclaration précompilée SQL qui permet une mise en cache de la requête et des métadonnées pour en accélérer l'exécution.

```

1  <?php
2  $hostname = 'localhost';
3  $username = 'username';
4  $password = 'password';
5
6  try {
7      $dbh = new PDO("mysql:host=$hostname;dbname=animals", $username, $password);
8      /** Définition des attributs du rapport d'erreurs **/
9      $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10     /** déclaration des variables **/
11     $animal_id = 6;
12     $animal_name = 'bruce';
13     /** Préparation de la requête SQL **/
14     $stmt = $dbh->prepare("SELECT * FROM animals WHERE animal_id = :animal_id AND animal_name = :animal_name");
15     /** Connection des variables aux paramètres de la requête **/
16     $stmt->bindParam(':animal_id', $animal_id, PDO::PARAM_INT);
17     $stmt->bindParam(':animal_name', $animal_name, PDO::PARAM_STR, 5);
18     /** Exécution de la requête **/
19     $stmt->execute();
20     /** Recherche du résultat **/
21     $result = $stmt->fetchAll();
22     /** Boucle pour afficher le résultat **/
23     foreach($result as $row) {
24         echo $row['animal_id'].'<br />';
25         echo $row['animal_type'].'<br />';
26         echo $row['animal_name'];
27     }
28     /** fermeture de la connection à la BD **/
29     $dbh = null;
30 }
31 catch(PDOException $e){
32     echo $e->getMessage();
33 }
34 ?>

```

Extrait de code 1 : Exemple de requête PDO

Ce système permet aussi de prévenir des injections SQL (cf. Extrait de code 1) par l'appel interne de la méthode *PDO::quote()*. Les injections SQL sont des actes malveillants qui consistent à

remplacer les données qui doivent être interprétées dans une requête SQL, par un code SQL qui modifiera la commande finale exécutée sur le serveur.

Je souhaitais présenter cette alternative qui n'a pas été sélectionnée dans mon projet car elle était très différente de l'existante. Cette autre solution pourrait être une solution avantageuse à mettre en place dans un projet ultérieur d'évolution de LabBook.

B. Sécurisation des accès

Travaillant sur l'environnement enseignant, une des exigences donnée était de ne pas anticiper des actes malveillants lors de l'utilisation de l'application. Le contrôle des données dans ce cadre était de veiller à ce que l'utilisation de caractères spéciaux ne génère pas de dysfonctionnement. J'ai dans un premier temps recensé les champs de la base susceptibles d'être mis à jour par des entrées provenant de l'interface utilisateur (cf. Tableau 2).

Tableau 2 : Champs de la base de données provenant d'entrées de l'utilisateur

TABLES	CHAMPS	TYPES	TAILLE
class	end_date	date	
class	class_name	varchar	32
comment	content	varchar	1000
group_u	group_name	varchar	32
labdoc	name	varchar	128
message	msg_content	varchar	1000
mission	code	varchar	16
mission	name	varchar	64
mission	description	varchar	1000
mission	assignment	varchar	1000
mission_conf	start_date	date	
mission_conf	end_date	date	
report_part	title	varchar	128
report_part	assignment	varchar	1000
ressource	name	varchar	64
ressource	ressource_path	varchar	255
teacher	teacher_domain	varchar	64
teacher	teacher_level	varchar	64
teacher	teacher_comment	varchar	
user	date_birth	date	
user	login	varchar	20
user	pwd	varchar	20
user	first_name	varchar	32
user	user_name	varchar	64
user	email	varchar	64
user	organization	varchar	64

Je souhaitais initialement repérer les types de données pour créer des fonctions qui auraient effectué les vérifications appropriées. Cette opération serait surtout utile dans le cas d'une refonte

de l'application pour mutualiser globalement ces traitements. J'ai repris les traitements précédemment utilisés à la demande de mon responsable, pour les intégrer dans mon code. Il a été décidé de n'appliquer que le minimum de contraintes sur les entrées des enseignants. Le travail était donc de s'assurer que les entrées des utilisateurs ne pouvaient pas altérer la stabilité du fonctionnement, sans anticiper d'actes malveillants.

Les **types de traitements** opérés sur les données sont variables :

- Contrôle d'entrée vide.
- Contrôle de taille maximale.
- Contrôle de syntaxe particulière.
- Contrôle de la présence de caractères non supportés.
- Filtre pour supprimer ou remplacer des caractères non souhaités.
- Application de traitements intégrés et automatisés :
 - JavaScript : *encodeURIComponent()*.
 - PHP : *mysqli_real_escape_string()*, *strip_tags()*.
- Application de traitements personnalisés.

Ces traitements dépendent du **type de données** :

- Date.
- Chaîne de caractères.
- Nombres.
- Code HTML.
- Code XML.
- Email.
- Adresse URL.

Et de la **provenance des données** :

- Valeur d'éléments du DOM.
- Éléments du DOM.

L'étude de ces différents paramètres est importante pour caractériser des processus de traitement distincts et réutilisables. Cette identification des données permet une meilleure stabilité et sécurité de l'application.

C. Conclusion

Le développement indépendant de la partie enseignante de l'application permet de la faire évoluer par étapes.

Les opérations sur les données auraient pu être mutualisées moyennant une réingénierie totale de l'application. Les modifications liées aux données sont critiques, que ce soit sur les tables, la gestion de leur accès ou des traitements sur les données. Ces éléments sont présents dans toute l'application et leur modification entraînent automatiquement des interventions coûteuses. C'est pour cette raison, que cela constitue une étape cruciale de l'élaboration d'une application.

III. Le CSS et les conventions de nommage

Les exigences données pour les noms de fichiers, de variables, de tables et de colonnes dans les tables, étaient l'utilisation de minuscules sans espaces avec comme séparateurs le caractère underscore. Pour les fonctions, la norme était de type « camel case / lower camel case », ce qui correspond à l'utilisation de minuscule pour le premier mot et de majuscules entre les mots suivants et pour les acronymes.

- lowerCamelCase
- UpperCamelCase

A. Le nommage dans l'architecture

Le nommage intervient dans différents éléments d'une application Web. Pour l'architecture globale, on peut réfléchir au nom des dossiers de l'arborescence pour faciliter la compréhension du système.

Dans les pages enseignantes, l'utilisation du nom de la page hôte pour le dossier et les pages liées (fonctions PHP, CSS, JavaScript, appel AJAX) a permis de l'utiliser en variable (cf. Extrait de code 2) et donc de rendre générique une partie du code.

```
<?php
// Reconstruction à partir du nom de la page, du nom du dossier et des autres pages liées
$page_name = substr(basename(__FILE__),0,-4);
$path_local = "/authoring_tools/".$page_name;
// Chargement de la page de bibliothèque de Fonctions liées à la page chargée
require($GLOBALS['dir_root'].$path_local.'/'.$page_name.'_fn.php');
?>
<script type="text/javascript">
    var global_path_local = "<?php echo $path_local ?>";
</script>
```

Extrait de code 2 : Mise en variable du nom de la page web

Durant ce stage, j'ai cherché à rationaliser les noms des fonctions pour plusieurs raisons. D'une part, certains outils dans les IDE proposent les fonctions en liste pour y accéder. Elles sont soit dans l'ordre d'apparition soit dans l'ordre alphabétique. On peut aussi les trouver par recherche. Dans ce cas, les premiers caractères vont être utiles pour le résultat proposé.

Une seconde raison pour nommer le nom des fonctions est l'établissement d'une règle commune qui permet d'harmoniser le code, sur laquelle le développeur se repose et fait gagner du temps dans des développements ultérieurs.

J'ai choisi durant le projet de commencer à normaliser le nom des fonctions (cf. Tableau 3). J'utilise en général l'élément qui est le sujet de l'opération pour débiter le nom de la fonction. S'ajoutent ensuite en général une action sur l'élément puis un état.

Cette démarche s'est effectuée au long de mon stage et certaines fonctions utilisent aussi un format où l'action comme *get* ou *put*, est placé au début du nom. Ces considérations n'ont pas d'impact sur le fonctionnement de l'application mais peuvent dans un premier temps homogénéiser son contenu.

Tableau 3 : Liste des fonctions contenues dans le fichier *gestio_classes.js*

<i>classAddCancel</i>	<i>getClass</i>	<i>learnerEditCancel</i>
<i>classAddValide</i>	<i>getClassTeachers</i>	<i>learnerEditValide</i>
<i>classDelete</i>	<i>getLearner</i>	<i>learnerSelect</i>
<i>classEdit</i>	<i>getTeacher</i>	<i>learnersImport</i>
<i>classEditCancel</i>	<i>importInfo</i>	<i>learnersImportCancel</i>
<i>classEditValide</i>	<i>importSelectAll</i>	<i>learnersImportValide</i>
<i>classListDisplay</i>	<i>importSelectFile</i>	<i>learnersListDisplay</i>
<i>classListFilterDate</i>	<i>infoAlert</i>	<i>learnerUnselect</i>
<i>classSelect</i>	<i>learnerAdd</i>	<i>reloadPage</i>
<i>filterInputDate</i>	<i>learnerAddCancel</i>	<i>teacherMove</i>
<i>filterInputEmail</i>	<i>learnerAddValide</i>	<i>teachersListDisplay</i>
<i>filterInputName</i>	<i>learnerDelete</i>	<i>updateClassLearners</i>
<i>filterInputString</i>	<i>learnerEdit</i>	

B. Le CSS (Cascading Style Sheets)

Le domaine pour lequel le choix du nommage est particulièrement important est celui des CSS. Le choix du nom des sélecteurs de *classes* et d'*identifiants* va définir la façon dont les CSS et le code JavaScript vont accéder aux éléments.

Avant d'aller plus loin dans les concepts liés au CSS, nous devons en présenter son rôle et son fonctionnement.

La mise en forme (couleurs, espaces, polices...) d'une page est définie par des paramètres liés aux éléments de cette page. Ces paramètres sont soit définis dans l'élément par l'attribut « style », soit dans une section de la page dédiée aux styles, soit, comme ce qui est fait dans la plupart du temps, dans une page « .css » spécifique. Les styles peuvent se propager sur des éléments « enfants » selon leurs définitions.

Le fichier de style est composé d'une suite de règles. Ces règles se composent d'un sélecteur qui identifie le ou les éléments cibles, suivis de l'ensemble des styles qui seront appliqués :

Sélecteur {attribut1: valeur1 ; attribut2: valeur2 ;}

Le sélecteur peut se composer de différents types d'éléments :

- Une **balise** HTML (input, select, body...).
- Une **classe** qui est représentée par un nom et sera affectée à une ou plusieurs balises de la page par l'attribut « class ».
- Un **identifiant** comparable à une classe, mais ne doit s'appliquer qu'à un seul élément de la page par l'attribut « id ».
- Une **suite de sélecteur** définissant une hiérarchie à respecter pour atteindre le sélecteur cible placé à la droite de cette liste.

Certaines règles d'application des styles sont importantes à signaler :

- Dans le cas de multiples règles appliquées à un élément au moyen de *classes*, l'ordre des règles qui vont être appliquées est celui de la feuille de style et non pas l'ordre des *classes* écrites dans la page HTML.

- Lorsque plusieurs éléments ancêtres définissent des valeurs différentes pour une même règle, les styles de l'élément le plus proche dans l'arborescence sont appliqués.
- Les sélecteurs ont un « poids », défini par le nombre et le type des éléments le composant. Ce « poids » va permettre aussi d'identifier le style à appliquer en cas de conflit.
- Quand une *classe* est rencontrée, pour trouver la règle à appliquer, le client va parcourir dans la feuille de style tous les sélecteurs se terminant par la classe donnée. Une vérification est faite dans le cas de sélecteurs composite, du chemin.

A partir de ces règles, il est recommandé de définir les styles des plus généraux aux plus particuliers sans utiliser de sélecteurs trop complexes.

C. Les méthodes d'utilisation du CSS

Les règles de fonctionnement des CSS ne sont pas nombreuses [CSS15], mais la façon de les utiliser de manière efficace demande beaucoup de temps. J'ai étudié au long de mon stage certaines méthodes. J'ai retenu trois méthodes distinctes et reconnues :

1. OOCSS

Une des pratiques est l'OOCSS initiée par Nicole Sullivan [stubbornella.org15] . Le concept d'OOCSS est de repérer des « objets CSS » qui se répètent, et de définir ainsi des classes réutilisables. Cette technique met en avant deux principes :

- Le principe de séparation de la structure et de l'apparence.
- Le principe de séparation du conteneur et du contenu.

Ce système est intéressant mais valable pour de gros sites où les règles sont effectivement répétées à l'identique. Dans le cas de sites où des variations s'appliquent dans la plupart des blocs, le bénéfice est moins grand et remet en cause l'utilisation de cette méthode.

2. BEM

Une autre technique est BEM (Block, Element, Modifier) [BEM15], qui part du concept qu'une page ou une application web peut toujours être rangée dans une arborescence de blocs, d'éléments et de modificateurs.

- Un **bloc** est une entité indépendante qui forme son propre contexte autonome (cf. Figure 34).
- Un **élément** est une partie d'un bloc. Le contexte d'un élément est celui du bloc (cf. Figure 35).
- Un **modificateur** est une propriété qui sert à créer des variantes sur un bloc ou un élément.

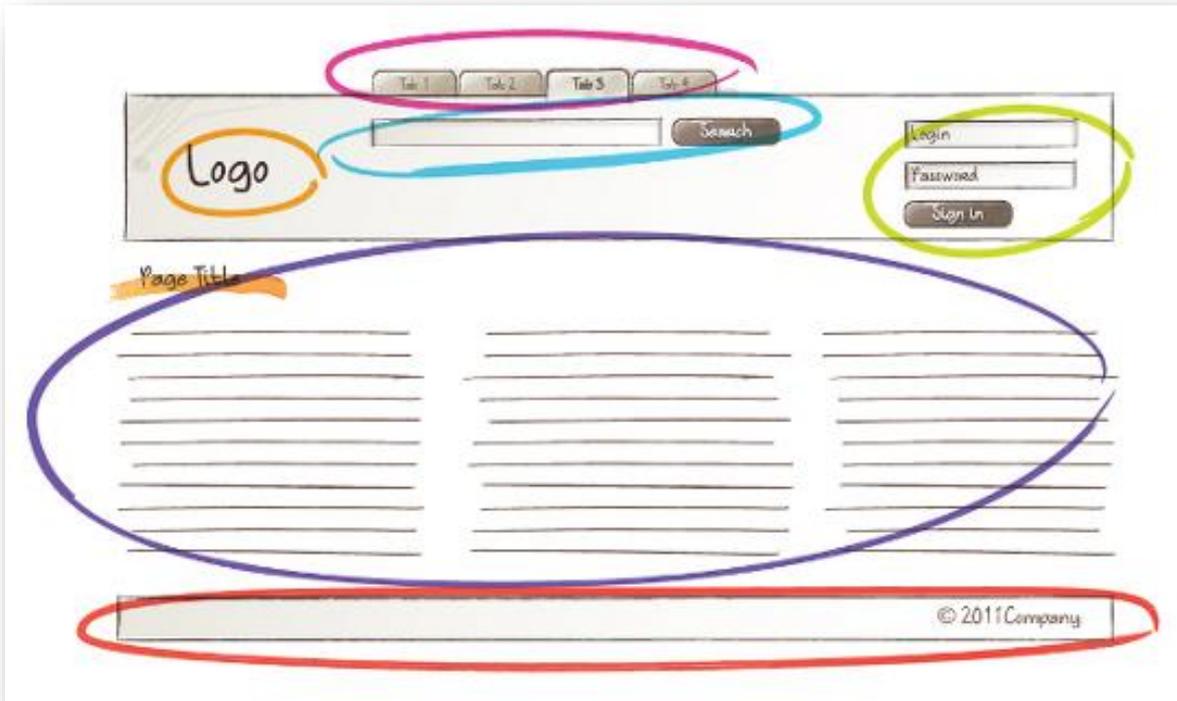


Figure 34: Exemple de blocs BEM [source bem.info]

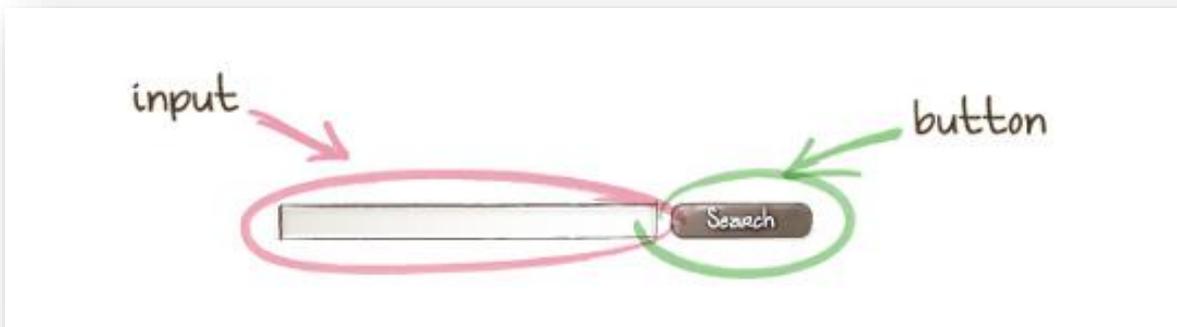


Figure 35 : Exemple d'éléments BEM [source bem.info]

Plusieurs styles de nommage sont mis en place à partir de ce concept :

- `.mon-bloc__mon-element--mon-modificateur`
- `.MonBloc-monElement--monModificateur`

3. SMACSS

La dernière méthodologie présentée ici est SMACSS (Scalable and Modular Architecture for CSS) proposée par Jonathan Snook. Cette méthode repose sur la distinction de cinq catégories :

- **Base** : Il s'agit des règles établies par défaut sur tout le site sur des éléments HTML basiques. Les réinitialisations CSS font partie de cette catégorie.
- **Layout** : Il correspond au découpage de l'architecture du site en sections majeures comme l'entête ou une barre latérale. Dans cette catégorie, les identifiants sont tolérés.

- **Module** : Comme son nom l'indique, il s'agit d'un bloc modulaire et réutilisable. Toute personnalisation doit passer par l'utilisation d'un modificateur. Le nom des différentes *classes* devront partager la même racine correspondant au module.
- **State** : Il s'agit ici d'un état, dans lequel va être placé un élément. L'attribut « important ! » peut être utilisé car l'élément n'est censé avoir qu'un état (cas d'un onglet actif ou inactif). La convention fera commencer la class par « *is-* ».
- **Theme** : On a ici des *classes*, souvent regroupées par fichier, qui vont redéfinir les styles de *base*, de *layout*, de *module* ou même de *state* en indiquant de manière claire quelles sont les règles qui définissent le thème et quelles sont celles qui n'en font pas partie.

D. Les problématiques rencontrées

Les méthodologies présentées précédemment illustrent la complexité que peut représenter la gestion des CSS avec ses choix de nommage. Le besoin des sélecteurs pour le code JavaScript s'ajoute à cette complexité. Bien que je ne l'aie pas mis en pratique, il est conseillé d'utiliser des *classes* et *identifiants* propres à ses besoins. Pour une utilisation qui ne relève pas de l'apparence, mais d'un besoin dans l'écriture du code JavaScript, il est préférable par exemple, de préfixer ces *classes* dédiées par « *js-* ». Ceci permet de modifier le style de la page et ses *classes*, sans altérer le fonctionnement du code.

Lors de la création de la page d'édition des missions, j'ai dû charger les outils de création de LabDoc. Le premier rendu a révélé les défauts mutuels de ma page et de l'outil COPEX en particulier. La page modifiait des éléments HTML de base, ce qui générerait des défauts d'affichage de l'outil. A cela s'ajoutait le fait que l'outil attendait les paramètres d'affichage qui n'étaient pas dans ma page.

Ce problème a mis en valeur l'importance de la manière dont sont mises en place les règles CSS et les conséquences sur la modularité et sur les opérations de maintenance. J'ai résolu ce problème en modifiant le CSS de ma page. La réingénierie du CSS est une opération lourde, d'autant plus si ses *classes* sont utilisées par le code JavaScript.

Le travail sur les pages de gestion des enseignants effectué dans la deuxième partie de mon stage m'a permis de travailler sur la réutilisation des règles CSS.

Nous avons choisi d'utiliser un format de table pour chacune de ces pages. Les tables ayant un entête avec de possibles icônes, des entêtes de colonne et optionnellement des filtres sur ces colonnes. À partir de ces éléments, j'ai créé des règles communes (cf. Extrait de code 3) pour ces pages qui ont non seulement allégé les feuilles de style de chaque page, mais aussi simplifié les modifications qui touchaient ces règles.

```

/*****
/* COMMUN - TABLEAUX ----- */
/* Les tableaux de liste sont composés de trois parties incluses dans une DIV qui permet de les cibler :*/
/* Un entête optionnel .bloc-header qui est une DIV */
/* Un tableau .table-header qui contient : */
/* - une ligne pour les titres .table-title, qui font l'objet du tri */
/* - une ligne optionnelle pour les filtres .table-filter, sur le contenu de la colonne */
/* Un tableau .table-list, qui contient la liste d'éléments */

.bloc-header { width: auto; height: 30px; padding: 5px 10px; font-size: 18px; vertical-align: middle;
  color: #fff; background-color: #a0a0a0;}
.bloc-header > div { display: inline-block;}
.bloc-header .align-right { margin-left: 10px;}
.bloc-header .fa { margin-left: 4px; color: #fff; cursor: pointer;}
.table-base { table-layout: fixed; border-spacing: 0;}
.table-base td { overflow: hidden; word-wrap: break-word;}
.table-header td { text-align: center; padding: 3px 5px;}
.table-title td { font-size: 16px; color: #fff; background-color: #a0a0a0; cursor: pointer;}
.table-title .fa { margin-left: 4px; color: #ffffff;}
.table-filter td { color: #ffffff; background-color: #a0a0a0; border-top: thin solid #bababa;}
.table-filter select { color: #ffffff; width: 100%; height: 100%; border: none; overflow: hidden;
  font-size: 12px; background-color: #a0a0a0; cursor: pointer;}
.table-filter option { overflow: hidden; font-size: 14px; color: #000000; background-color: #ffffff;}
.table-list { display: inline-block;}
.table-list tr:nth-child(odd) { color: #4b4b4b; background-color: #efefef;}
.table-list tr:nth-child(even) { color: #4b4b4b; background-color: #cfcfcf;}
.edit .table-list tr:nth-child(odd) { color: #4b4b4b; background-color: #ffffff;}
.edit .table-list tr:nth-child(even) { color: #4b4b4b; background-color: #cfcfcf;}
.table-list tr:hover td { color: #ffffff; background-color: #a0a0a0;}
.table-list td { padding: 3px 5px; font-size: 14px; overflow: hidden}
.table-list .fa { color: inherit; cursor: pointer;}
.td-selected { color: #cfcfcf; background-color: #555555;}

```

Extrait de code 3 : Règles communes de style pour les pages de gestion enseignantes

E. Conclusion

Il n'y a pas de méthode assurant un résultat optimal dans le domaine du nommage et des CSS. Il est nécessaire de s'intéresser aux différentes techniques, pour se les approprier et pouvoir les utiliser au mieux dans le contexte. La simplicité apparente de ce type de travail est trompeuse, et la taille des applications ainsi que le travail de maintenance sur celles-ci poussent à investir du temps pour mieux maîtriser les techniques.

J'ai personnellement trouvé l'approche SMACSS intéressante car souple. Elle peut donc être utilisée comme base pour fusionner les méthodes existantes en leur prenant les principes les plus intéressants :

- Des éléments peu ou pas modulaires de type conteneur ou structurel.
- Des éléments modulaires, réutilisables.
- Une définition hiérarchique des éléments (ex : `class='btn btn-module1'`).
- Le concept d'état par défaut et d'états modifiés.
- La séparation de la structure et de l'apparence.
- La séparation du conteneur et du contenu.

Une fois les types d'éléments déterminés et les règles d'utilisation définies, il est alors possible d'établir une règle de nommage adaptée.

La formalisation de l'utilisation des CSS est de ce fait un exercice qui nécessite une analyse pointue, par la logique et la rationalité de l'organisation choisie, et son efficacité une fois mise en pratique.

En effectuant ce travail sur les CSS, j'ai mis en pratique les concepts de module réutilisable et de bloc. Une utilisation plus fine et optimisée des CSS demande un investissement important car dès que l'on commence à modifier le style dans son architecture toute la page est impactée.

IV. Architecture « Authoring »

L'élaboration des pages s'est effectuée en deux étapes principales. La première étant la conception de la page d'édition des missions dans l'environnement enseignant initial. C'est aussi lors de cette étape que s'est faite l'acquisition des technologies nécessaires au codage, et mes premières propositions et décisions face aux problématiques d'architecture, de nommage et de programmation.

La deuxième étape est la refonte de l'interface enseignante, et l'intégration de la page précédemment créée dans ce nouveau contexte.

A. Exigences et contraintes

Après une création rapide de maquette, ma première proposition lors de l'élaboration de la page d'édition des missions a été rejetée. J'utilisais alors le gestionnaire d'événements pour déclencher des processus. Cette méthode était peu utilisée sur les pages du site qui utilisaient des appels explicites aux fonctions par des événements « *onClick* » ou « *onChange* ». Dans un contexte de maintenance, il est plus aisé de comprendre les séquences et enchaînements d'événements quand ceux-ci sont visibles directement dans le code HTML de la page.

Dans mes choix, j'ai intégré cette exigence de simplicité de compréhension et de maintenance tout en utilisant le plus possible les méthodes et techniques déjà en place.

Un certain nombre d'exigences étaient déjà formulées :

- Respecter au mieux la sémantique du contenu.
- Ne pas anticiper des besoins et contraintes non encore exprimés.
- Éviter la redondance.
- Commenter le code pour en faciliter la maintenance.
- Appeler les fonctions JavaScript, par les « *events* » HTML *onXXX* depuis le DOM.
- Utiliser JQuery pour simplifier l'écriture du code et améliorer la compatibilité entre clients web.
- Limiter le nombre de requêtes AJAX au minimum.
- Ne paramétrer les requêtes AJAX en mode synchrone qu'en cas d'obligation.
- Ne tester les arguments des fonctions que si nécessaire car travaillant sur l'environnement enseignant les risques sont moindres.
- Veiller à ce que les valeurs de retour des fonctions soient utiles.
- Utiliser le style procédural, pour l'écriture du code.

Une autre contrainte était d'impacter le moins possible l'application en place, d'où l'utilisation d'un dossier « *authoring_tools* » pour y placer tous les composants de l'environnement spécifique aux enseignants.

B. Choix de l'architecture logicielle

Tout au long de mon projet, j'ai souhaité formaliser mon travail, mes choix d'organisation et de technique pour homogénéiser le rendu des travaux. J'ai aussi veillé à m'assurer de l'accord de mon responsable sur les démarches que j'entreprenais, telles que les choix d'organisation des fichiers.

Après avoir étudié l'architecture existante, et ne pouvant y mêler mes fichiers, j'ai décidé de garder la logique employée sur une partie de l'application, et sur les composants de messagerie et commentaires. Elle consiste en le stockage des fichiers d'une page dans un même dossier, portant le nom de la page (cf. Figure 36). Les noms des fichiers se déclinent toujours de la même manière :

Nom_de_la_page.php : Page principale qui charge les bandeaux d'entête et de pied de page ainsi que tous les composants nécessaires. Elle ne contient pas de traitement, l'objectif étant d'y trouver la structure HTML de la page.

Nom_de_la_page_fn.php : Elle contient les fonctions PHP spécifiques à la page, qu'elles soient appelées au chargement initial ou via le fichier « ****_AJAX.php* » qui reçoit les requêtes AJAX.

Nom_de_la_page_AJAX.php : Ce fichier réceptionne les requêtes AJAX et appelle les fonctions choisies via le fichier « ****_fn.php* ».

Nom_de_la_page.css : Contient toutes les règles CSS de la page.

Nom_de_la_page.js : Contient tout le code JavaScript de la page.

Toutes les pages de l'enseignant peuvent utiliser des fonctions communes, disponibles dans les fichiers « *authoring* ». Les autres fichiers « *data_processing.** » et « *auth_cnx_db.php* » étant respectivement dédiés, comme leur noms l'indique, à des traitements génériques sur les données et aux opérations basiques sur la base de données.

Je n'ai pas placé les pages principales dans leurs dossiers respectifs mais à la racine du dossier « *authoring_tools* » pour palier à une conséquence du chargement des outils de LabDoc dans la page d'édition des missions. Dans ces outils, de multiples chemins sont référencés en relatif, et pour garder le tout fonctionnel sans modifier l'existant, il fallait que ma page soit exactement dans un sous-dossier à la racine du site.

Les fichiers réceptionnant les requêtes AJAX auraient pu être fusionnés avec les bibliothèques de fonction « ***_fn.php* » mais nous les avons conservés pour garder notre modèle proche de l'existant.

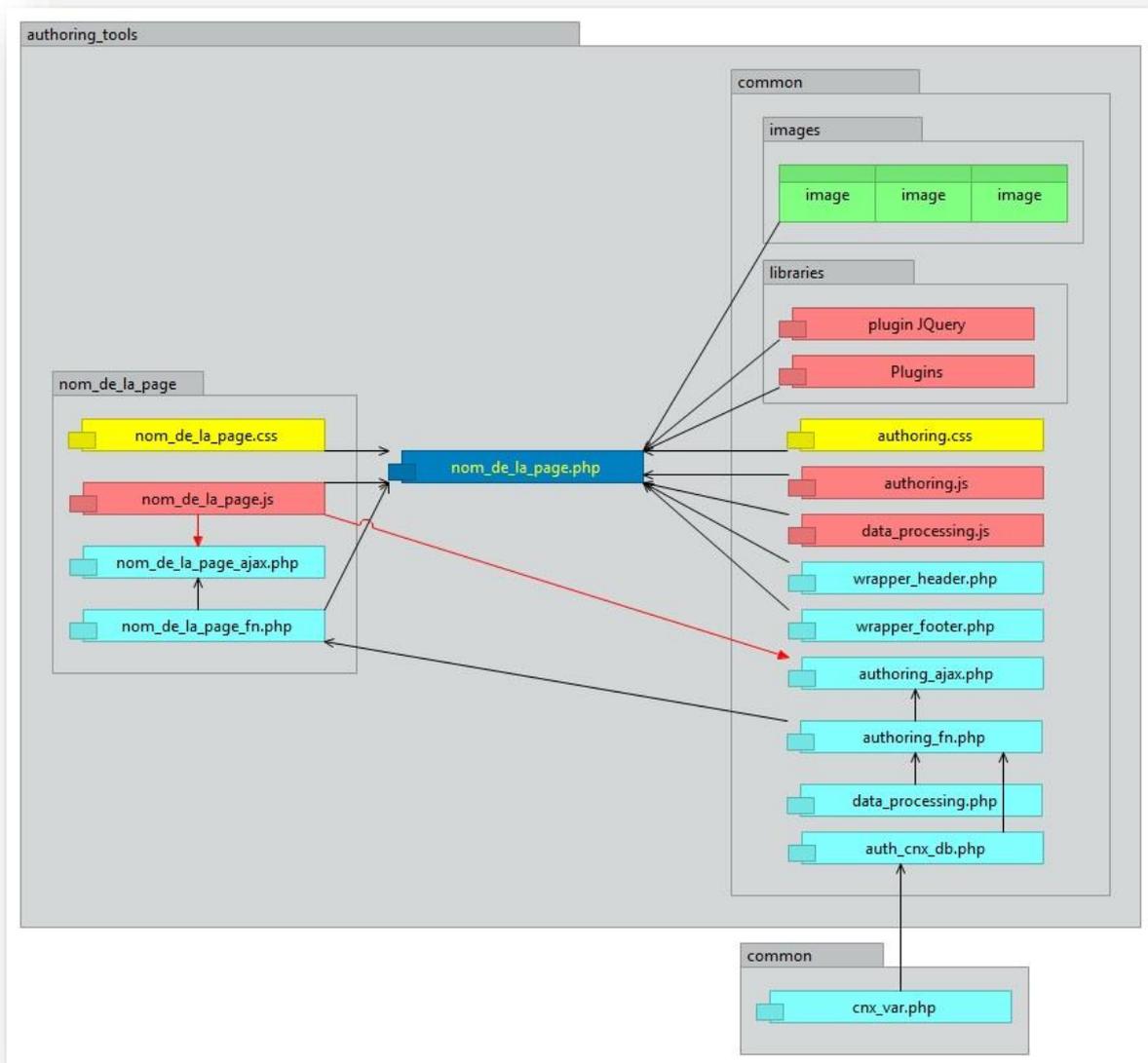


Figure 36: Architecture d'une page de l'interface des enseignants.

Dans celui-ci, je n'ai pas essayé d'organiser le code pour différencier des sections complètes ou des fichiers de type « Modèle », « Vue » ou « Contrôleur ». L'architecture globale de l'application n'avait pas été bâtie dans ce sens, et ma priorité était de faciliter la prise en main et la maintenance. J'ai donc organisé le contenu des bibliothèques de fonctions JavaScript et PHP par thème, point sur lequel je reviens dans la section suivante. Les notions MVC restent cependant importantes. J'ai donc, par exemple au sein d'une même fonction, séparé les traitements et les lignes de code qui préparent la Vue. Cela facilite la visibilité et la compréhension du code HTML.

C. Conception de la page d'édition des missions

Cette page est la première qui a été conçue. Elle a fait l'objet de plusieurs mises à jour pour, dans un premier temps, répondre aux requêtes faites lors des tests utilisateurs, et dans un deuxième temps, pour l'intégrer dans la nouvelle interface dédiée aux enseignants.

Son fonctionnement est particulier, et proche de la page d'édition des rapports de l'étudiant. Ceci est dû au fait qu'elle charge les outils d'édition de LabDoc, et accède à la majorité des tables de la base de données. Une autre exigence qui conditionne tout son fonctionnement est que la plupart des données doivent être enregistrées à la volée, pour permettre à l'enseignant d'interrompre quand il le souhaite son travail sans perte de données.

Cette exigence conduit à mettre en place la page fixe qui va se mettre à jour ainsi que la base de données par l'utilisation de requêtes AJAX.

L'utilisation d'AJAX change la manière d'organiser les pages. Dans des schémas standards, une page se charge avec les données préalablement sélectionnées. Ensuite des traitements pour organiser les données peuvent être effectués localement suite à des événements sur le poste client. S'il y a modification des données ou nécessité de les mettre à jour sur le client, la page doit se recharger sur le poste client ; elle doit rétablir le contexte si nécessaire. Dans notre cas, la page est chargée une seule fois. Si je souhaite mettre à jour des données dans la base, une requête AJAX est envoyée au serveur et des scripts PHP effectuent les modifications. De même pour une mise à jour de la page sur le client après la requête AJAX, le serveur envoie soit des données et le client s'occupe de la Vue, soit le serveur envoie le code HTML directement prêt à être intégré dans la Vue.

Le travail sur cette page a été très intéressant car elle fait appel à de nombreuses techniques comme le téléchargement de fichiers (dans les deux directions, client-serveur), des menus contextuels, le chargement d'outils extérieurs et globalement beaucoup de travail avec JavaScript pour gérer les nombreux éléments.

D. Conception des pages enseignantes

Les pages des enseignants sont constituées d'ensembles de données présentées de manière tabulaire. La demande de mon responsable était de pouvoir effectuer de multiples tris et filtres sur ces données. Les quelques formulaires de mise à jour des tables n'étaient pas très complexes. Un des choix de conception était de gérer les traitements sur l'affichage des données localement ou par des rafraichissements de pages. Pour limiter les requêtes sur le serveur et les transferts réseaux, j'ai entrepris d'utiliser la même technique dans chacune de ces pages.

Des fonctions PHP renvoient des objets JSON (cf. Extrait de code 4), reçus dans des variables JavaScript (cf. Extrait de code 5) au chargement de la page principale. La page a donc, dans des variables objet, toutes les données dont elle a besoin. Le code JavaScript s'occupe alors d'effectuer au besoin des traitements pour ré-agencer, filtrer ou trier ces données et met à jour la page.

Par cette méthode, la page n'est pas rafraichie tant que les données n'ont pas été modifiées. Une conséquence de cette technique, est que le code PHP ne s'occupe plus que du Modèle dans un schéma MVC. Le Contrôleur et la Vue étant pris en charge par le code JavaScript.

L'état de la page, l'élément sélectionné, le tri et les filtres actifs, sont stockés dans un tableau (*global_page_state*) qui est utilisé pour le rechargement de la page.

Ces méthodes, comme toutes les autres (gestion des filtres, des tris) sont reprises à l'identique dans chacune des pages afin de faciliter la prise en main et la maintenance. Une modification ne devient qu'une opération à répéter sur chaque page.

```
/**
 * Renvoie un objet JSON des la liste des enseignants avec leurs infos
 * @return bool|string
 * @author A. Le Meillour - 16/08/2015
 * @last_modif
 */
function getListTeachers(){
    $teachers_arr = array();
    $sql = 'SELECT DISTINCT u.id_user, u.user_name, u.first_name, u.email, u.organization,
                t.teacher_domain, t.teacher_level, t.teacher_comment
        FROM user as u, teacher as t
        WHERE t.id_teacher = u.id_user AND
                t.id_teacher <> \'1\'
        ORDER BY u.user_name ASC';
    $result = dbQuery($sql);
    if (mysqli_num_rows($result) > 0) {
        $teachers_arr = dbFetchAssocAll($result);
        return dbRealEscapeString( json_encode($teachers_arr) );
    } else {
        return false;
    }
}
```

Extrait de code 4 : Préparation des données au format JSON pour les variables JavaScript

```
// contient toutes les données nécessaires à la page
var global_data = JSON.parse('{
    "root_classes":<?php echo getListClass($id_user);?>,
    "root_learners":<?php echo getListLearners($id_user);?>,
    "root_teachers":<?php echo getListTeachers();?>
}');
```

Extrait de code 5 : Création de l'objet JavaScript à partir des données PHP au format JSON

Le travail de mutualisation des opérations pouvait peut être poussé plus avant, mais les pages auraient perdu en maniabilité et en simplicité.

E. Conclusion

Cela a été une chance de pouvoir travailler sur l'ensemble des pages enseignantes pour mettre en place une architecture de plusieurs pages, et la mutualisation de certains de leurs éléments. Cela m'a aussi permis de réfléchir sur l'homogénéité de leurs interfaces et de techniques de codage employées.

Lors du travail sur les requêtes AJAX, j'ai observé combien les traitements asynchrones pouvaient provoquer des effets indésirables s'ils n'étaient pas maîtrisés.

J'ai aussi intégré les bénéfices du travail avec des objets JSON, qui facilitent l'échange de collections d'objets entre le client et le serveur.

V. Conclusion de la deuxième partie.

Le contexte des applications Web est particulièrement intéressant et complexe. Les techniques sont multiples, elles évoluent très vite. À cela doit s'ajouter une expertise dans les pratiques à employer pour utiliser ces techniques le plus efficacement possible.

Ce sont ces pratiques et méthodes qui sont les plus ardues à acquérir dans l'étendue des informations disponibles.

L'utilisation de cadriciel pour la structure et la gestion de l'application comme Angular JS ou pour la présentation comme Bootstrap, permet de s'affranchir de certaines problématiques. LabBook, qui autant que possible évite les dépendances de ces cadriciel, offre l'exercice de la réflexion sur les différentes méthodes à employer.

Tous les sujets étudiés dans cette partie confirment l'importance de la modularité. Les calendriers des projets ne permettent pas d'investir toutes les technologies et leurs évolutions avec expertise. Il faut donc gérer le changement, et plus l'application est structurée et maîtrisée, plus les modifications pourront aussi être maîtrisées.

L'enjeu des évolutions techniques entreprises dans mon stage a été de garder un code simple et maintenable tout en cherchant à le structurer. J'ai mis en œuvre la nouvelle interface pour les enseignants tout en travaillant sur ses aspects techniques. Mon travail s'inscrit dans celui de mes prédécesseurs tout en intégrant les contraintes d'évolution et certaines techniques comme l'utilisation du format JSON.

TROISIEME PARTIE - PROBLEMATIQUES ORGANISATIONNELLES

Un projet de développement d'application Web ne s'arrête pas à la seule définition des besoins et l'écriture du code pour y répondre. Ces activités sont essentielles car directement liées au cœur du projet. Mais à la manière dont un projet nécessite des ressources pour sa propre gestion, il est important de souligner l'importance de problématiques connexes.

Dans cette partie, je présenterai tout d'abord les outils que j'ai utilisés, avec les bénéfices et critiques que j'ai pu leur attribuer. Ensuite, j'exposerai les différents types d'activités effectuées pendant mon projet et les questions autour de l'arbitrage du temps attribué à chacune. Dans les deux chapitres suivants, j'expliquerai les méthodes utilisées pour la spécification et la documentation dans le projet LabBook au vu du contexte et des exigences.

I. Les outils utilisés

La gestion d'un projet de développement WEB requiert des outils divers selon l'activité ; de la communication au codage, en passant par la spécification. Ces outils peuvent être standards (comme la messagerie) ou spécialisés et complexes (comme un IDE). Le gain de productivité tiré par l'utilisateur dépend de l'outil choisi et des moyens mis en œuvre pour l'utiliser au mieux. Le temps est une contrainte importante dans le choix des applications que j'ai utilisées, il faut trouver et avoir un résultat rapide, donc une prise en main facile. J'ai utilisé premièrement les outils disponibles dans mon environnement comme Microsoft PowerPoint. Un autre vecteur a été les sources d'informations consultées dans ma démarche d'auto-formation. J'ai visionné plusieurs vidéos provenant de site WEB comme Grafikart [Grafikart15] ou Video2Brain, qui m'ont permis de découvrir les éditeurs de code que j'ai utilisés pendant mon stage. Je présente ci-dessous une liste non exhaustive des logiciels de mon stage, accompagnée de quelques commentaires.

A. Le maquetage

Très rapidement, il a fallu formaliser sous forme de maquettes d'interfaces les objectifs du stage. Les premières esquisses ont été faites au crayon pour ensuite être formalisées avec Powerpoint 2013 (cf. Figure 37). Les objets disponibles dans ce logiciel, auxquels s'ajoutent les images téléchargeables sur les moteurs de recherche, permettent avec les fonctionnalités d'alignement, de produire rapidement des schémas d'écran efficaces. Il existe d'autres outils comme Visio proposant de multiples objets graphiques, mais les objets sont forcement limités et définis, et les règles de gestion et de positionnement contraignent leur utilisation. La solution avec PowerPoint et des images choisies a été la plus efficace pour représenter mes propositions.

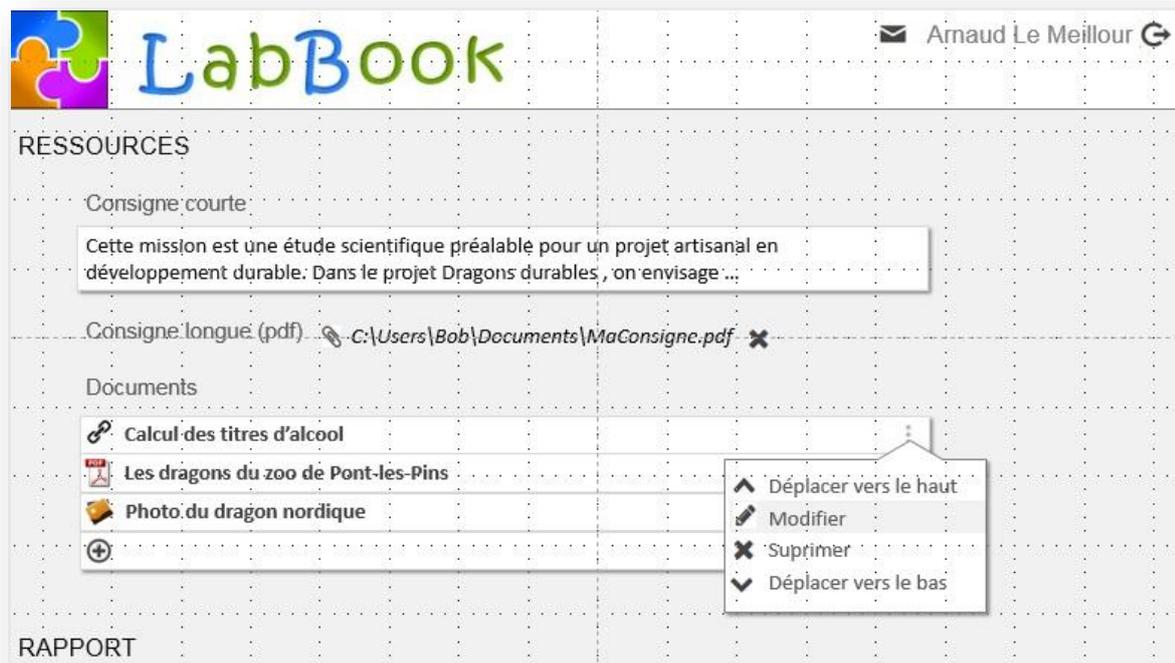


Figure 37: maquette sous PowerPoint 2013

B. La programmation

Dans mon activité de développement, j'ai utilisé deux types d'outils principaux. En premier lieu, les éditeurs de code, qui dans mon cas avaient la particularité de ne pas avoir à gérer de compilation puisque JavaScript et PHP sont des langages interprétés. Dans un second temps, pour les phases de correction d'erreurs et d'ajustement des styles, j'ai utilisé les outils de développement intégrés aux navigateurs WEB.

1. Les éditeurs de code

Du simple éditeur de texte au logiciel intégré comportant de nombreuses fonctionnalités pour améliorer la productivité du développeur, il n'est pas simple d'arrêter un choix définitif.

Durant mon stage, j'ai testé quatre outils : **Notepad ++ 6.8.3** ; **Sublime Text 3** ; **Aptana Studio 3** et **PHPStorm 8.0.3**.

Les deux premiers (**Notepad ++ 6.8.3** ; **Sublime Text 3**) sont des éditeurs légers utilisant quelque dizaine de méga octets de mémoire, dont les fonctionnalités peuvent être augmentées par l'ajout de modules. Notepad++ [Notepad++15] est un éditeur fonctionnant sous MS Windows qui évolue grâce à une communauté active. Les fonctionnalités sont limitées mais l'outil est réactif et gratuit avec une licence GPL [GNU15]. Dans un registre similaire, Sublime Text [Sublime Text15] est quant à lui payant, et disponible sous MS Windows, Linux et OS X. Les fonctionnalités intégrées comme la multi-Edition [SEVILLEJA10] ou la gestion de projet sont très riches et il possède aussi un système de gestionnaire de paquets téléchargeables [Package Control15] qui offre la possibilité d'ajouter certaines fonctionnalités similaires à celles offertes dans les EDI. J'ai installé quelques extensions gérant l'alignement et la proposition de méthodes natives JQuery ou JavaScript.

Aptana Studio et **PHPStorm** sont, quant à eux, de complets EDI dédiés au développement web.

La gestion de l'indentation et la coloration syntaxique sont des fonctionnalités basiques offertes par tous ces outils mais lors de l'utilisation de PHPStorm, mon outil principal, j'ai pu découvrir des fonctionnalités intéressantes qui simplifiaient mes actions.

Je présente ci- dessous quelques fonctionnalités utiles :

- Gestion d'un site distant : Permet la mise à jour des fichiers d'un serveur web distant par différents protocoles, dans mon cas SFTP (Secured File Transfer Protocol, c'est à dire transfert de fichiers via un canal crypté).
- Affichage de la structure du fichier sélectionné : Propose un accès direct aux fonctions JavaScript et PHP appelées dans le fichier, avec quelques opérations contextuelles au clic droit. Cela facilite l'accès au code dans les gros fichiers. On retrouve ici l'importance des choix de nommage pour identifier et rassembler les fonctions par thème.
- « Find usages » : Fonctionnalité très utile qui permet de lister les portions de code, dans les différents fichiers, où est située la fonction sélectionnée. Elle offre plus d'options que les simples recherches de chaînes de caractères habituelles. Associée à la commande (Ctrl + clic) sur le nom d'une fonction, qui ouvre le fichier hôte de celle-ci et l'affiche, cette fonctionnalité m'a permis lors de réingénierie de mieux comprendre et simplifier des enchainements entre fonctions.

Sans faire une liste exhaustive de toutes les commandes et fonctionnalités utilisées comme le contrôle avec la base de données ou celui de la syntaxe, on peut dire que l'outil PHPStorm, en tant que EDI, fournit toute une palette de services améliorant la productivité. Il faut souligner cependant que les EDI réclament des ressources matérielles plus importantes, il faut compter sur une consommation de plusieurs centaines de méga-octets lors de leurs utilisations. C'est pourquoi si toutes les fonctionnalités ne sont pas nécessaires, l'utilisation d'éditeurs plus simples peut suffire.

Lors des activités de codage, j'ai utilisé régulièrement les deux types d'éditeurs en parallèle.

2. Les outils de développement intégrés aux navigateurs

Les exigences qui m'ont été données pour le projet LabBook étaient le support des plateformes Google Chrome et Mozilla Firefox. J'ai néanmoins été amené à faire quelques tests sur Apple Safari car je développe l'interface pour les enseignants qui utilisent souvent des matériels Apple. Plusieurs anomalies ont été remontées et même si cette plateforme n'était pas officiellement supportée, des correctifs ont pu être apportés. J'ai donc à cette occasion utilisé l'outil « *web inspector* » de safari mais j'ai cependant principalement travaillé avec le « *Developer Tools* » de Chrome version 47.0.2514.0 et en second, l'outil de développement intégré au navigateur Firefox version 40.0.3.

Ces outils pour les développeurs sont des applications complexes qui offrent comme les EDI de puissantes fonctionnalités. Ceux-ci ne s'arrêtent pas à donner les informations nécessaires à la correction des erreurs. Ils aident le développeur, par des informations comme l'obsolescence de fonctions ou leur utilisation inappropriée. Leurs outils permettant de mieux comprendre le fonctionnement de l'application dans son contexte et d'en affiner les propriétés.

Nous retrouvons des outils communs sur les navigateurs Chrome et Firefox, pour visualiser le code HTML, éditer les styles, déboguer le code JavaScript, voir l'état des variables etc. J'ai utilisé une partie des fonctionnalités, qui peuvent être abordées sous plusieurs angles. Dans le cas de la gestion des styles, l'outil permet de tester rapidement des propriétés. On peut aussi voir cette utilisation

de manière plus qualitative, en observant la multiplicité des styles, et leur propagation. Cette approche est aussi utile dans l’outil « réseaux » qui donne des informations sur les délais et temps de réponse.

J’ai principalement utilisé les outils de développements fournis par Chrome pour leur richesse. Ceux-ci fournissent un émulateur de client mobile simulant des bandes passantes dégradées. Chrome permet aussi un audit de la page web signalant des anomalies et proposant des optimisations (cf. Figure 38).

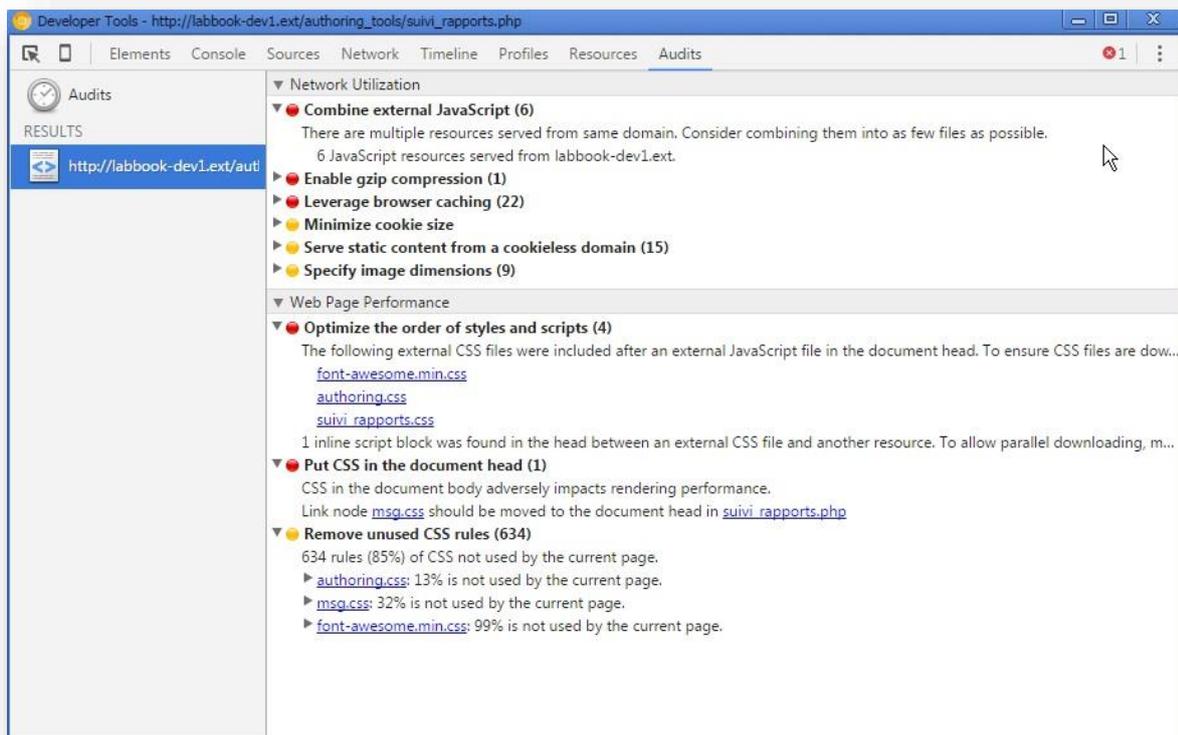


Figure 38: Outil d’audit du navigateur Google Chrome

C. La modélisation

Lors de mon projet sur l’application LabBook, j’ai eu besoin de modéliser différents éléments comme les données, ou le séquençement d’actions.

Pour la représentation des données, j’ai utilisé le logiciel WhiteStar UML pour le diagramme de classes qui était déjà en place ainsi que MySQL Workbench qui m’a permis de générer son diagramme **EER** (Enhanced entity–relationship) (cf. Annexe E), et d’accéder à la base distante.

Dans mon travail sur l’outil Copex, j’ai eu le besoin de schématiser les fichiers et fonctions ainsi que le séquençement de certaines opérations. Après avoir essayé plusieurs applications comme MS Visio ou MS Excel, j’ai finalement décidé d’utiliser le logiciel Archimate [The Open Group15] que j’ai découvert dans mon unité d’enseignement NFE210 au CNAM. Le nombre d’objets ainsi que de relations disponible dans ce logiciel est important, de plus un objet créé est unique et disponible dans plusieurs schémas. Cela permet de créer rapidement de nouveaux schémas sans avoir à recréer systématiquement tous les éléments. Pour mon besoin, j’ai formalisé mes fichiers et les fonctions qui leur sont intégrées par des composants d’application Archimate dans une vue

de structure d'application. J'ai utilisé des relations de flux pour représenter le séquençement entre les fonctions dont le nom était constitué de lettres pour distinguer les chemins alternatifs, et des chiffres pour ordonner les liens extérieurs séquentiels (cf. Annexe G). Comme tout élément de documentation, se pose la problématique de la fiabilité du contenu dans le temps. Néanmoins, ces schémas m'ont semblé utiles car dans le contexte de maintenance de l'application par mon responsable ou par d'autres stagiaires, ceux-ci peuvent être un outil efficace pour faciliter la compréhension de l'architecture de l'application.

D. Conclusion

Le besoin de productivité valide l'importance du choix des outils que l'on va utiliser dans les différentes phases de son projet. Il faut y ajouter les bénéfices dans la qualité de travail que l'on peut en retirer et le meilleur rendu. Ce projet m'a permis de découvrir des outils, mais il est nécessaire de rester ouvert et à l'écoute de produits différents et innovants pour pouvoir s'en enrichir au besoin. Je continue à approfondir mes connaissances sur ces outils pour me les approprier pleinement et développer progressivement une culture sur ce domaine.

II. La gestion du projet

Selon l'Organisation Mondiale de Normalisation, « *Un projet est un processus unique qui consiste en un ensemble d'activités coordonnées et maîtrisées, comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques, incluant des contraintes de délais, de coûts et de ressources.* » [ISO03].

Selon les différents critères qui identifient un projet, mon stage ne s'apparente pas à cette définition au sens strict du terme. Je vais aborder dans ce chapitre certains des critères qui identifient un projet, ce qui donnera une grille d'analyse sur le contenu de mon stage.

A. Qualités, coût et délais (QCD)

Un projet est parfois symbolisé par le triangle reliant ses trois composants que sont **son coût**, **son délai** et **sa qualité**. Dans le cadre ce stage subventionné par un financement extérieur, les seules ressources génératrices de coût sont les moyens matériels mis à disposition et dans mon cas, le temps de mon responsable, Cédric D'HAM dédié à l'encadrement de ce stage. L'accent est donc surtout posé sur les délais, pour faire avancer l'application et résoudre les erreurs critiques. L'application LabBook est utilisée par des enseignants, et génère des données. La priorité a donc été de conserver l'intégrité des données et la disponibilité du service.

B. Les personnes et les rôles

Dans l'équipe MeTAH du LIG, quatre personnes (Cédric D'HAM, Isabelle GIRAULT, Claire WAJEMAN et Patricia MARZIN-JANVIER) travaillent occasionnellement sur l'application en testant les nouvelles versions, et décident des modifications et évolutions. D'autres enseignants extérieurs l'utilisent et participent à son développement moins directement par des signalements d'erreurs et des suggestions. Mon responsable, Mr D'HAM, maintient les développements effectués par les stagiaires et lui-même et il est plus particulièrement en charge des aspects techniques. Dans le cadre du fonctionnement du LIG, des stagiaires sont aussi régulièrement encadrés pour faire avancer certains projets. Provenant souvent de DUT, Master 1 ou Master 2, ces stages d'une durée de 2 à 4 mois sont ciblés sur des portions de l'application.

Dans ce contexte, mon rôle a été particulier, compte tenu de la durée du stage sur neuf mois, et du peu de connaissance en programmation que je possédais au début de mon stage. Mon expérience professionnelle en informatique système et l'enseignement suivi dans mon cursus au CNAM m'ont cependant permis de progresser rapidement dans les aspects techniques et d'avoir une vision plus globale des problématiques rencontrées.

J'ai bénéficié d'une grande autonomie, n'ayant principalement que mon responsable comme interlocuteur. Sa principale exigence était de préserver le fonctionnement et la maintenabilité de l'application.

C. Le périmètre du projet et les livrables attendus

Définir un périmètre pour un projet nécessite d'identifier l'impact sur tous les modules de l'application [GestionDeProjet15]. Dans le cas de LabBook, le développement successif de parties par des étudiants différents au long des années, rend complexe la maîtrise de l'application de par son manque d'homogénéité. L'architecture pourrait s'apparenter à un élément organique dans sa complexité et son évolution. Le périmètre devient donc difficile à évaluer et le niveau d'incertitude augmente avec l'hétérogénéité des développements effectués.

Je reprends ci-dessous la liste des livrables attendus, présentée dans mon dépôt de sujet de mémoire d'ingénieur accompagné de commentaires.

1. Module de création des missions et de pré-structuration des rapports

La première réalisation demandée a été la page de création et d'édition d'une mission pour l'enseignant. L'attente de cette fonctionnalité était forte, et cette réalisation intégrerait moins de dépendance vis-à-vis des autres composants de l'application. Elle représentait un bon exercice dans mon processus d'auto-formation sur les techniques web. Le développement de cette page s'est avéré plus complexe que prévu car, comme celle de création de rapports, elle touche la plupart des tables de la base de données et charge les outils de création de LabDocs. Ceux-ci n'avaient pas tous été initialement étudiés pour être chargés dans d'autres pages et certains effets de bord ont été découverts.

Mon travail d'auto-formation a été complexe car au-delà des technologies à intégrer, se sont posées de multiples problématiques liées aux méthodes, aux « *bonnes pratiques* » ou à l'architecture logicielle.

2. Module de suivi du travail des étudiants

Le module enseignant a été complètement repensé. Les enseignants utilisant l'application avaient fait part de leurs difficultés pour la prise en main. Les interfaces pour l'étudiant et l'enseignant étaient fusionnées sur la page de choix de missions. Nous avons donc décidé avec mon responsable de revoir complètement l'application pour l'accès enseignant, ce qui a modifié le contenu et le planning établi mais était indispensable.

3. Sécurité des connexions

Les contraintes du projet n'ont pas permis certaines améliorations en termes de sécurité. Ces améliorations auraient pu toucher l'accès aux données en utilisant l'interface PHP Data Object (PDO), et plus globalement en revoyant l'architecture de l'application. Cependant, ces modifications auraient engendré une réécriture trop importante de l'application, ce qui ne pouvait pas être effectuée durant mon stage. De plus, les serveurs Web sont hébergés et gérés par l'équipe MISI du LIG et il n'était pas prévu que je travaille sur la configuration des serveurs.

4. Amélioration des performances et de la stabilité de la plateforme

J'ai travaillé particulièrement sur l'optimisation des pages créées durant mon stage. J'ai aussi étudié l'outil Copex qui présentait des problèmes de perte de données. Cela m'a permis d'améliorer certaines fonctions et requêtes AJAX. Un des problèmes récurrents est le traitement suite à des requêtes AJAX Asynchrones. Je préconiserais en plus de l'étude sur l'optimisation de certains modules, un diagnostic du contrôle des entrées utilisateurs et de leurs traitements pour assurer une meilleure stabilité.

5. Evolution des outils de production des LabDocs

J'ai pu dans l'environnement enseignant, faire évoluer l'outil des traitements de texte TinyMCE. Lors de mon étude de Copex, j'ai également étudié la mise à jour du composant logiciel JSTree, qui s'est avéré impossible sans perdre plusieurs fonctionnalités déjà en place. En effet, nous utilisons une méthode d'affectation du nœud en Html qui a été supprimée.

Les contraintes du projet n'ont pas permis d'aborder certains livrables tels que l'étude du service de commentaires, du service d'export des rapports ou encore du module de diagnostic et de retour lié à l'outil de protocole.

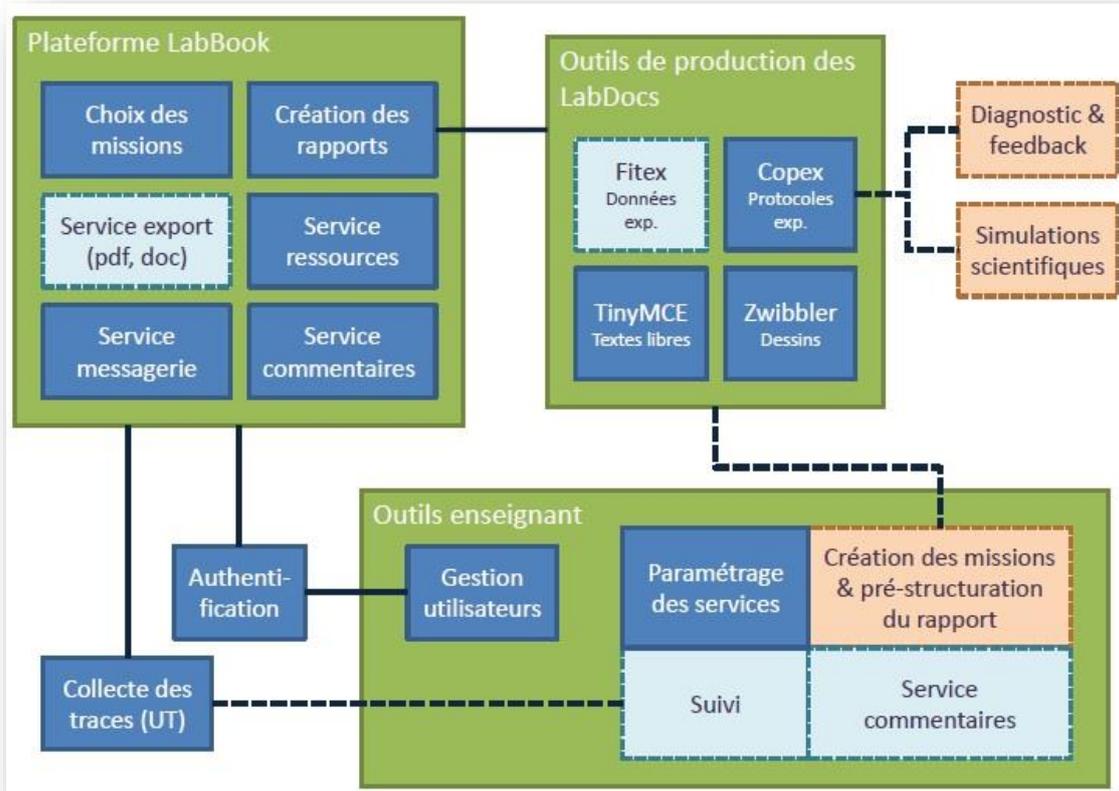


Figure 39 : Découpage modulaire de LabBook.

D. Le planning et le pilotage

Du fait de l'adaptation aux problèmes techniques rencontrés et à l'organisation adoptée, je n'ai pas suivi de planning strict durant mon stage. Celui-ci s'est déroulé suivant un cycle de développement qui s'apparente aux méthodes agiles [agilealliance15].

1. Itératif

La conception débute sur une demande, une observation de l'existant et la notification des exigences et contraintes. Suivait une étape de maquettage et d'échanges autour de celle-ci jusqu'à une base permettant la conception. Durant la conception, je sollicitais mon responsable au besoin pour clarifier certains aspects ou valider des propositions. Quand la conception était opérationnelle, nous la proposons en test à l'équipe MeTAH travaillant sur LabBook pour l'évaluer. Ensuite, nous organisons une réunion afin de relever les erreurs rencontrées lors des tests, et recevoir et valider les demandes de modifications. L'étape suivante consistait à produire le livrable dans l'environnement de test avant mise à jour en production. Sans que ce cycle de développement ne se reproduise complètement, des demandes de modifications donnaient lieu à sa reprise.

2. Incrémental

Le projet global d'interface enseignante n'a pas été élaboré et prévu au début du stage. Sa conception s'est déroulée en deux étapes, avec la page d'édition des missions et ensuite, quatre nouvelles pages auxquelles s'ajoutaient des modifications sur la première page créée. A ces créations de pages s'ajoutaient des modifications sur les pages existantes et le travail sur l'outil Copex. Le développement de l'application LabBook a donc été aussi incrémental bien que le développement de l'interface finale contenant les pages de gestion enseignante, se soit fait en une fois dans un souci d'harmonie et d'efficacité.

3. Adaptatif

Le déroulement du projet s'est construit en fonction des problématiques rencontrées et des priorités données par mon responsable. L'auto-formation, la gestion de la qualité, les problèmes d'effets de bord et le maintien fonctionnel de l'application ont orienté les directions et priorités durant le projet.

Le pilotage de projet comme LabBook est spécifique, car bien que de faible coût, toute modification engagée doit être finalisée pendant le stage. De ce fait, toucher des éléments qui impactent de manière trop importante l'application devient complexe et il est indispensable de maîtriser leur finalisation dans le temps imparti du stage.

E. Conclusion

Le déroulement et le contenu d'un projet dépendent de multiples paramètres sur lesquels nous pouvons plus ou moins influencer. L'enjeu de ce type de stage est de pouvoir produire une évolution utile tout en conservant la maintenabilité de l'application.

Dans mon rôle de stagiaire ingénieur, j'ai souhaité réfléchir autant sur les techniques que j'acquerrais que sur leur utilisation. Cela m'a poussé à aller au-delà du développement de fonctionnalités demandées, pour réfléchir aux problématiques qui se sont présentées à moi.

III. Les principales étapes

Le déroulement d'un stage de ce type se compose de multiples activités. On retrouvera sans doute dans la plupart des projets informatiques ces mêmes tâches, avec des criticités et temps alloués variables selon le type de projet, l'expérience des personnes et les méthodes de projets choisies.

Je présente dans les sections suivantes, les activités dont l'importance est significative en temps. Je ne traiterai donc pas des réunions et du maquettage, bien qu'importants, n'ont pas représenté beaucoup de temps.

A. La rétro-conception

La documentation de LabBook se compose principalement de schémas et des rapports de stage de mes prédécesseurs. Pour comprendre les aspects fonctionnels, j'ai profité des explications verbales de mon responsable de stage, et pour les éléments plus techniques, j'ai puisé dans les rapports et le code source.

Les rapports expliquaient des points particuliers qui m'ont été, dans un premier temps, difficile à intégrer sans description globale de l'architecture. Ceci est sans doute dû au fait que les stages effectués par mes prédécesseurs, compte tenu de leur courte durée, ne concernaient qu'une partie très restreinte de l'application.

Le code lui aussi est empreint des strates correspondant aux stages qui ont fait évoluer LabBook. Le code porte la trace de la mise en place, dans le temps, de règles, de techniques et d'organisation ainsi que les travaux de chacun des stagiaires. On peut constater dans mon propre code, mon évolution durant le stage.

Dans cette partie sur la rétro-conception, l'accent est mis sur l'importance de la qualité du code, pour la maintenance et aussi pour guider les personnes qui interviennent par la suite. On travaille plus sur du code existant que sur de nouveau développement, de ce fait, les efforts sur l'architecture, les commentaires, les techniques et les bonnes pratiques participent à la formation du programmeur.

Plusieurs courants existent au sujet de la documentation dans, ou hors du code. Le fait est que celui-ci reste le dernier maillon pour comprendre le fonctionnement de l'application.

Il est possible d'intervenir sur de petites parties sans comprendre les interactions et séquences liées à certaines actions. Il devient alors compliqué d'assimiler l'application avec du recul et on perd la maîtrise de celle-ci.

Mon travail de rétro-conception dans LabBook a donc été d'intégrer et d'utiliser les techniques et bonnes pratiques que j'identifiais et que je maîtrisais. À partir de celles-ci et d'autres techniques issues de mon auto-formation, j'ai pu proposer à mon responsable des pratiques pour les formaliser.

B. L'auto-formation

L'auto-formation a été un élément capital de mon stage, car je n'avais ni la culture d'un programmeur ni les connaissances dans les technologies WEB. A part ma formation initiale en DUT, j'ai, au cours de mon travail d'administrateur système Windows, effectué des scripts en PERL et VBScript. Mes connaissances web quant à elles, viennent de l'étude de mon UE NFE102 au CNAM.

La variété de techniques à assimiler simultanément ainsi que leurs évolutions rapides dans le temps représente une barrière d'accès à ce type de connaissance.

HTML, CSS, JavaScript, AJAX, PHP, MYSQL et toutes les techniques liées comme JQuery demandent beaucoup d'investissement. De plus, pour chacune de ces techniques, il y a un long

chemin pour acquérir les bonnes pratiques et les différentes méthodes en profondeur. Heureusement, internet propose de multiples sources d'information ou tutoriaux à qui veut s'en donner la peine.

En tant qu'étudiant ingénieur je ne voulais pas me contenter d'apprendre la syntaxe des langages et enregistrer des lignes de code pour exécuter des opérations. Je souhaitais m'attarder sur les bonnes pratiques et avoir un regard sur la qualité du code.

Le temps d'auto-formation a été particulièrement important au début du stage. J'ai continué à me former sur de nouvelles techniques par la suite, tout en faisant avancer le projet. Une autre conséquence de l'acquisition de nouvelles connaissances est la volonté de les intégrer dans son code. On se retrouve alors avec des problématiques d'hétérogénéité du code ou de temps alloué pour la réingénierie logicielle afin d'atténuer ce dernier phénomène. A plusieurs reprises, j'aurais souhaité reprendre des parties de code déjà fonctionnelles ou retravailler sur les CSS. Mais dès que j'évaluais une durée trop importante ou inconnue pour effectuer ce genre de travail, je préférais souvent mettre de côté ces tâches.

Cela dit, malgré la complexité de l'auto-formation nécessaire à cette mission, j'ai pu, grâce à l'acquisition des connaissances nouvelles, à mon expérience en scripts et à mon recul, mettre à profit cette auto-formation tout au long de l'élaboration du projet.

C. La programmation

La programmation fait ici état des périodes hors réingénierie. Elle est globalement la suite des deux phases précédentes d'auto-formation et de rétro-conception. J'ai adopté la démarche tout au long de ce stage, de formaliser et structurer mon travail de programmation. En cherchant à homogénéiser et utiliser de bonnes pratiques pour l'architecture logicielle, les techniques de codage ou les commentaires, l'enjeu était multiple. Je travaillais à la fois sur la qualité du code et les effets positifs que l'on pouvait en retirer, et aussi sur un cadre sur lequel je pouvais m'appuyer, pour me concentrer sur d'autres problématiques.

Le travail de programmation est un exercice particulier. La plupart du temps, il existe, de nombreuses manières d'effectuer une opération. Dans un cadre autre que celui d'un stage, le coût est un facteur important et il est nécessaire de produire rapidement des fonctionnalités dans l'application. La liberté de formulation et de technique peut être de ce fait restreinte. Après s'être accordé avec mon responsable sur des exigences en termes de codage, j'ai eu la liberté de proposer les techniques que je trouvais efficaces aux vues de ces contraintes. J'ai de cette manière utilisé une police d'icônes pour faciliter la gestion des alignements, couleurs et chemins. J'ai aussi travaillé avec le format JSON pour simplifier le transfert de collection de données.

D. La réingénierie logicielle

Comme l'auto-formation, la réingénierie est un investissement, un coût d'exploitation des ressources présentes, pour un bénéfice attendu dans le futur.

Devant mon constat sur l'évolution de LabBook par le biais de stages, et la demande de mon responsable d'impacter le moins possible l'environnement étudiant « déjà fonctionnel », j'ai particulièrement travaillé sur l'homogénéité du code et sa maintenance.

Dans cette démarche, au fur et à mesure de mes progrès, j'ai essayé de revenir sur le code déjà écrit pour l'améliorer. Le projet avançant et avec lui le volume de code, certaines modifications généralisées comme l'optimisation des CSS étaient difficiles à entreprendre.

J'ai développé parallèlement les pages principales de l'enseignant, pour là aussi homogénéiser l'ergonomie, et le code. Cela m'a permis aussi de faire des économies d'échelle sur l'opération de réingénierie car dans le cas d'une évolution comme le traitement sur les tableaux, la modification était à répéter à l'identique sur les autres pages.

Le bénéfice attendu ici est aussi que les personnes ayant à reprendre le code s'approprient facilement les techniques et l'architecture par sa redondance.

Comme expliqué dans le chapitre précédent (cf. Chap. II.D), notre méthode de développement était aussi génératrice de réingénierie pour intégrer les modifications demandées.

E. Conclusion

J'ai traité précédemment les activités pour lesquelles j'ai investi du temps qui répondaient à la priorité de mon responsable, soit l'aspect technique et la réponse à ses demandes d'améliorations. Il a fallu trouver des compromis entre les phases d'auto-formation, de mise en pratique, de création, d'amélioration de l'existant tout en gardant une application en production. Tout ceci a été très enrichissant et possible grâce à un fort investissement. Il est important de maintenir des phases d'auto-formation et de réingénierie pour progresser, suivre l'évolution des techniques, des pratiques et améliorer la qualité.

IV. La gestion de l'information

Dans ce chapitre, je vais aborder les aspects de mon stage liés à la communication et en particulier aux différents types de documentations. Bien que l'accent n'ait pas été mis sur cette thématique pendant le projet, je vais dans les sections suivantes présenter le travail effectué du fait de l'importance du sujet.

A. La communication

Dans cette section, par communication, je traite des échanges professionnels avec mes différents interlocuteurs durant mon stage.

La grande majorité de ces échanges a été avec mon responsable Cédric D'HAM, Maître de conférences et enseignant chercheur en didactique dans le domaine de la chimie. Du fait de son activité, il était régulièrement hors des bureaux du laboratoire MeTAH. Nous n'avons pas eu de réunions régulièrement mais nous échangeons par email et nous nous rencontrons au besoin pour des explications ou pour se mettre d'accord sur des choix techniques ou fonctionnels.

J'ai également échangé avec Isabelle GIRAULT, Claire WAJEMAN et Patricia MARZIN-JANVIER lors des deux réunions concernant la validation de la page d'édition des missions après quatre mois de travail, et l'ensemble des pages enseignantes, un mois avant la fin du stage. Elles ont été chargées de tester les pages, de rapporter les erreurs rencontrées et de proposer des modifications. Ces réunions ont permis d'avoir une vision utilisateur et de décider des dernières modifications à apporter avant mes tests finaux et la mise en production.

B. Les documents de définition

Les documents de définition permettent d'échanger et de formaliser l'information.

Dans un projet de développement, ils sont obligatoires dès que l'on se trouve dans la cadre d'une prestation commerciale. C'est le cas avec l'outil de dessin Zwibbler de LabBook, commandé à la société Hanov Solution Inc. sur la base de leur outil générique, pour laquelle un document de spécification répertorie de manière exhaustive les fonctionnalités demandées.

Le reste de l'application a été développé en interne, car c'est un outil expérimental et il n'a pas pour but d'être déployé à grande échelle. Les ressources pour son développement sont limitées et expliquent l'appel à des stagiaires, ainsi que la priorité mise au développement de nouvelles fonctionnalités.

Il n'y a pas de document formalisant de manière exhaustive l'expression des besoins à remplir par LabBook et ses outils qui sont le cœur de l'application. La création d'un tel document demanderait beaucoup de temps, et s'y ajouterait le problème de toute documentation, qu'est son maintien à jour.

Les spécifications se composent donc de documents reprenant certains points techniques, de schémas et majoritairement de maquettes commentées (cf. Figure 40).

Affichage du protocole

Modifier, dupliquer, supprimer le labDoc

Le <label> en gris italique

En gris italique si c'est le <defaultContent>, en noir plain si c'est le <content>

Si il ya un commentaire il s'affiche en dessous (gris plain pour <comment> et gris italique pour <defaultComment>)

Protocole expérimental

Question de recherche ou objectif
Décrivez l'objectif de votre expérimentation : la question à laquelle vous voulez répondre et/ou les objets que vous voulez produire

Hypothèses ou résultats attendus
Lister les hypothèses que vous souhaitez tester au cours de votre expérimentation et/ou les résultats que vous pensez obtenir.

Principe de la manipulation
Décrivez rapidement la stratégie, les moyens que vous allez mettre en place. Le principe de manipulation ressemble à un mode opératoire succinct ne contenant pas les paramètres de la manipulation.

Liste du matériel
fiolle jaugée de 10 mL (3)

Mode opératoire.

Voici la première étape de la procédure
... que je commente
Réaliser cette étape 4 fois :

	1	2	3	4
Paires de pattes	2	3	2	3
Fruit donné en repas	banane	banane	pomme	pomme

Voici la première action de l'étape
... que je commente aussi

Voici la seconde action de l'étape

Voici une sous-étape
Totalement vide !

Deuxième étape de la procédure
Sans commentaire

Ceci est une action structurée à titre imposé : A partir d'ici j'ai mis ce que je voulais :)
Apparemment j'ai modifié le commentaire

Ceci est une action structurée à contenu paramétré : Il faut acheter 5 pommes et petites bananes.
Oui, ça n'est pas du bon français...

Ceci est une action structurée à contenu typé : Je fais une action avec un paramètre libre toto + une grandeur avec unité 145.7E-9 m + du matériel issu de la liste des matériels utilisés fiolle jaugée de 10 mL (2). Cette action produit un nouveau matériel dont le nom est spécifié par l'étudiant ici : sol-toto

La liste du matériel s'affiche sur 4 colonnes et par ordre alphabétique. Entre parenthèses le <nb> (si nb=1, on n'affiche rien). On n'affiche ni les <type>, ni la <description>, ni le <comment>.

Le <title> d'une action structurée en italique

Les <param> d'une action structurée en gras

Figure 40: Exemple d'écran, de la maquette de l'outil COPEX

Les spécifications que j'ai produites ont suivi les mêmes méthodes, par l'élaboration de maquettes. J'ai tenté de créer des documents de spécifications pour la page d'édition des missions (cf. Annexe 0) et plus globalement sur mes pages enseignantes, mais la méthode de développement que nous avons employée, itérative et incrémentale, obligeait à modifier régulièrement ces documents. Il aurait été utile d'avoir des documents de synthèse, avec des spécifications fonctionnelles pour les outils d'élaboration de protocole, de donnée et de dessin, car ils répondent à des demandes spécifiques du groupe MeTAH. Ce type de document demanderait moins de mises à jour car il ne nécessite pas de rentrer dans le détail de l'implémentation, mais au vu du temps disponible, cela n'a pas été mis en place. Néanmoins, lorsque c'était nécessaire, mon responsable me fournissait les explications permettant d'avoir une vue plus fonctionnelle pour mettre du sens derrière les éléments techniques et mieux les appréhender.

A. Les documents de réalisation et de suivi.

Pour répondre à la problématique de maintien à jour des informations, j'ai choisi d'enrichir les commentaires dans le code, conforté par le fait qu'il s'agissait également d'une demande initiale au projet.

Là aussi, j'ai étudié les bonnes pratiques et règles que l'on pouvait trouver pour, en plus des commentaires de ligne placés derrière des doubles barres obliques, formaliser les commentaires de fonction (cf. Figure 41).

```
/**
 * Effectue un tri si nécessaire sur l'objet JSON : global_data_missions
 * Mise à jour de l'affichage de la liste des missions dans une table
 * @param column_name - colonne sur laquelle s'effectue le tri
 * @param loop [true|false] indique si l'on boucle sur l'ordre du tri ou pas
 * @author A. Le Meillour - 16/08/2015
 * @last_modif A. Le Meillour - 10/09/2015
 */
function missionListDisplay(column_name, loop) {
    var obj_mission = global_data_missions;
```

Figure 41: Commentaire de fonction

L'utilisation de format normalisé a de multiples avantages.

Premièrement, dans mon outil de programmation PHPStorm, le squelette du commentaire est généré automatiquement en reprenant les variables disponibles. Outre cette facilité de création, certains éditeurs de code permettent de fournir des informations contextuelles sur les fonctions. Dans PHPStorm, en utilisant le raccourci Ctrl+Q sur une fonction sélectionnée, une fenêtre renvoie les informations provenant du commentaire normalisé (cf. Figure 42).

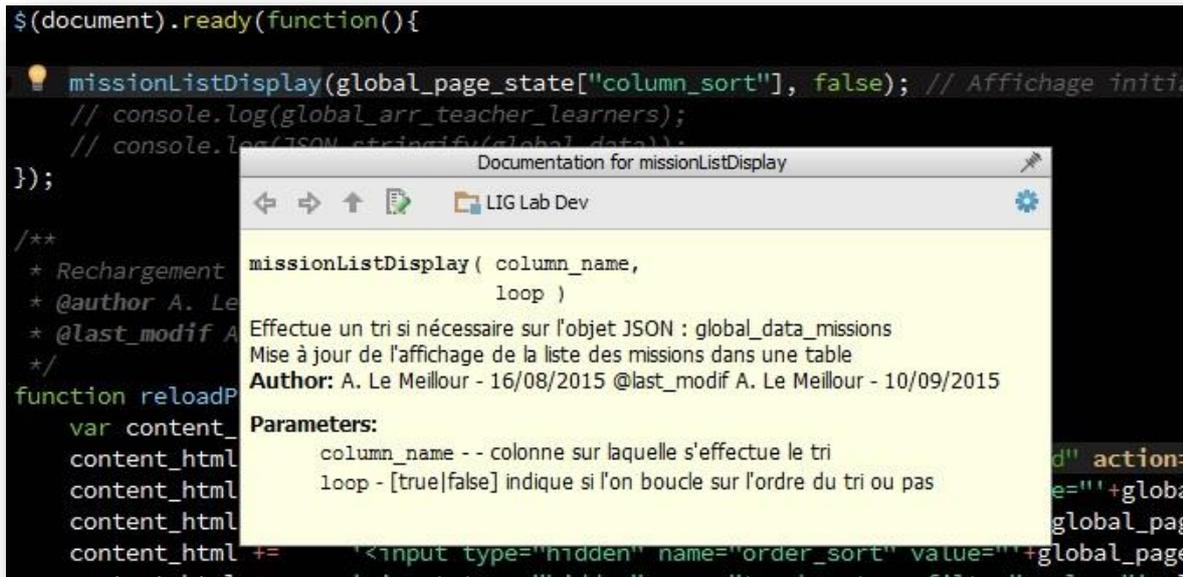


Figure 42: Information contextuelle sur une fonction, depuis un commentaire normalisé

Un autre avantage, est la possibilité d'utiliser un générateur de documentation automatique du type Doxygen [Doxygen15]. Celui-ci crée une documentation (cf. Figure 43) sous plusieurs formats dont l'HTML, et permet d'accéder plus rapidement aux informations renseignées dans le code (cf. Figure 44).

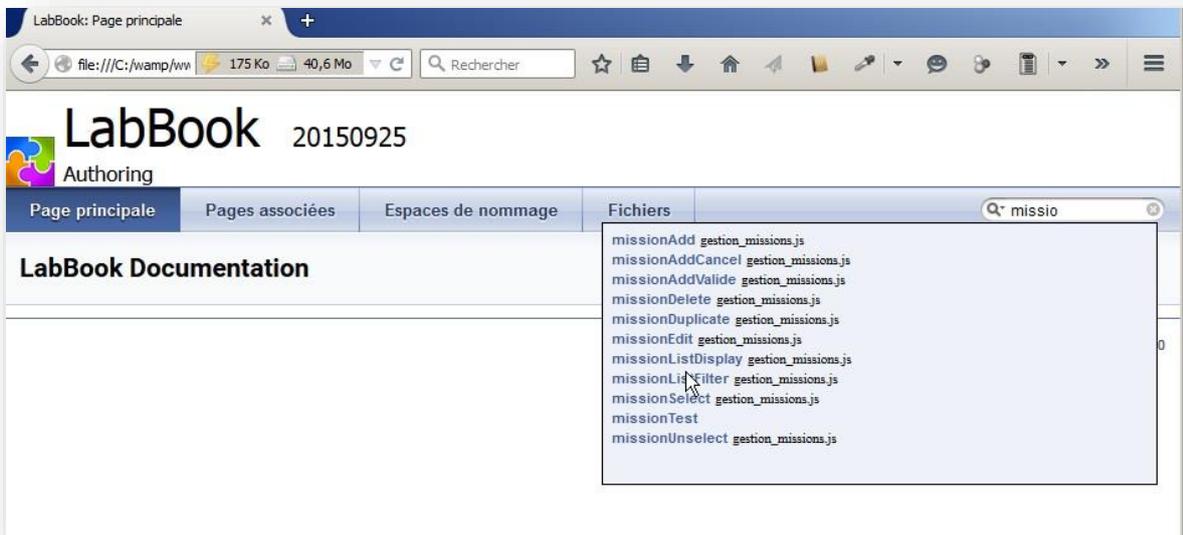


Figure 43: Interface d'accueil générée par le logiciel Doxygen



Figure 44: Information sur une fonction dans le site généré par Doxygen

Pour faciliter la maintenance, j'ai ajouté dans les commentaires des informations sur l'auteur et la dernière mise à jour. Nous n'avons pas à notre disposition de gestion de version de type GIT ou SVN. Mon responsable, Cédric D'HAM, ne souhaitait pas introduire une infrastructure de gestion de version pour une application de faible taille et pour laquelle peu de personnes interviennent simultanément, tout au plus lui et un stagiaire.

Les risques étaient minimisés car seul mon responsable avait accès au site de production qu'il mettait à jour. De plus, je n'avais accès en écriture que sur mes pages et je passais par lui pour toute modification sur des fichiers extérieurs.

J'ai également tenté de formaliser les tests sur les pages enseignantes (cf. Annexe H). Je souhaitais produire une liste exhaustive des éléments à valider et attester des contrôles effectués. J'ai décidé de ne pas continuer cette tâche car vérifier les réactions et événements sur tous les éléments des pages, en essayant toutes les possibilités sur les données (en longueur de chaîne, type de caractères) représentait une quantité de travail sans rapport avec le temps donné sur les autres activités. Les tests se sont arrêtés à l'évaluation par l'équipe, ma revue du code et mes propres tests.

B. Les rapports

Une dernière source d'information correspond aux différents rapports rendus à la fin des stages de mes prédécesseurs. Mon responsable me les avait communiqués avec les autres documentations car ils décrivent des parties de l'application aux dates des stages et permettent de comprendre certaines problématiques rencontrées dans le passé. Ils renvoient souvent à des points techniques particuliers car les stagiaires ont une demande ciblée.

Mon mémoire fera partie de cette documentation, et c'est dans cette optique que j'ai souhaité y traiter de problématiques de fond, d'axe d'analyses multiples, pour ne pas recenser uniquement la mise en place d'une série de solutions à des problèmes de programmation.

C. Conclusion

L'information sous ses différentes formes, tels que des comptes rendus, des rapports ou des documents de spécifications a vocation à garder une trace en vue d'une utilisation possible. Le temps nécessaire à sa création et sa maintenance vient pondérer sa justification. A cela s'ajoute la

problématique de son format, et il peut être intéressant de connaître une information dans son contexte temporel. Dans des projets comme LabBook qui évoluent perpétuellement, le code, les techniques, les spécifications, tout est à placer dans son contexte. La multiplicité des supports et de leur nature variée vient aussi complexifier l'accès à l'information.

Des réflexions peuvent être menées pour décider des documents nécessaires, ainsi que de leur structure et de la formalisation de leurs contenus. On rencontre ce genre d'approche plus facilement dans des entités de tailles importantes qui industrialisent leurs processus.

Dans une application comme LabBook, c'est l'anticipation de son utilisation pour les prochaines années qui peut piloter la décision de son investissement nécessaire pour aux tâches de documentation. C'est l'arbitrage entre la complexité grandissante de l'application qui va requérir plus d'informations pour la maîtriser, et le coût à investir pour produire une documentation utile.

V. Conclusion de la troisième partie.

Les différents thèmes abordés dans cette partie ne relèvent pas de l'activité principale de programmation, demandée pendant mon stage. Ils permettent de mieux comprendre le contexte et la variété des problématiques rencontrées. Le temps a souvent été un facteur limitant de mes investigations et c'est dans le temps et avec les expériences que pourront s'affiner mes choix et mes méthodes de travail.

Ces activités connexes font pleinement partie du projet et c'est pourquoi les questions et enjeux s'y attachant méritent d'être exposés car les bénéfices à retirer d'améliorations touchent toutes les parties prenantes. Que ce soit en qualité de travail, en productivité ou en maîtrise du projet, cerner le contexte et pouvoir apporter les réponses appropriées, requiert de s'attacher à tous ces détails.

Les cours théoriques sur les projets, les différents types d'organisations et méthodes peuvent maintenant faire écho à cette expérience.

CONCLUSION

Les recherches dans lesquelles s'inscrit le projet LabBook obligent l'applicatif à évoluer au long des années, en s'adaptant aux requêtes qui lui sont faites. L'exercice devient compliqué pour faire évoluer ce produit par le biais de stage d'étudiants, dans un environnement technologique web en perpétuel évolution.

A. Le bilan du projet

LabBook souffrait d'une interface enseignante confuse et certains composants comme Copex manquaient de fiabilité. Mon travail a été autant de faire évoluer l'application que d'améliorer sa fiabilité. L'interface des enseignants est à présent finalisée, que ce soit en termes de fonctionnalités que d'ergonomie. De plus, sa dissociation de l'interface des étudiants permet des évolutions simplifiées de chacune. Mon apport a touché aussi la fiabilité de l'application que ce soit dans le fonctionnement des outils de LabDoc, ou dans l'architecture mise en place. En effet, plus l'architecture est réfléchiée et rigoureuse, plus sa maintenance et son évolution sont aisées. J'ai effectué ces tâches dans le cadre des exigences de simplicité qui m'était demandé et sans altérer l'existant.

B. Perspectives et évolutions

LabBook est dès à présent confié à un nouveau stagiaire en charge de fiabiliser l'outil Fitex. J'espère que mon travail et mon souci de cohérence dans l'architecture s'inscriront avec celui de mes successeurs. LabBook peut évoluer dans ses fonctionnalités ou son ergonomie, mais il reste de multiples points d'amélioration de l'application qui sont à distinguer selon que soit choisie une démarche de continuité ou de rupture avec les éléments qui constituent le contexte et les contraintes de celle-ci.

Une fois les décisions métier établies, il est demandé à l'application d'être fiable et réactive. Ce sont les objectifs techniques principaux.

Actuellement, Cédric D'HAM doit gérer les aspects techniques et fonctionnels de LabBook en plus de ses activités d'enseignement et de recherche. Il n'y a en effet pas d'ingénieur développement disponible en permanence pour assurer la maintenance et l'évolution du produit. Les choix techniques découlent donc de cette contrainte : la conception et les technologies utilisées doivent faire l'objet d'une concertation avec Cédric D'HAM, car il difficilement envisageable qu'il doive se familiariser avec une architecture de code différente pour chaque intervenant.

L'aspect maintenabilité est donc prédominant dans les choix techniques concernant LabBook. Malgré les atouts qu'apporterait l'implémentation d'un modèle MVC, l'application est somme-toute de taille modeste, et une action ponctuelle dans le code ne pose pas de difficultés particulières.

Donc, les aspects que je soulèverais en termes d'évolutions techniques vont dans le sens d'une amélioration de la fiabilité. On peut citer :

- Une normalisation du nommage plus poussée.
- L'utilisation plus importante de patrons de conception tels que « les modules » ou « les modules révélateurs » qui permettent d'encapsuler des fonctions et propriétés.

Conclusion

- La gestion de l'utilisation par plusieurs enseignants d'une mission et plus globalement de l'édition d'éléments communs (classes).
- Une gestion plus modulaire et autonome des outils de création des LabDocs, par l'intégration dans chaque outil de la gestion de leurs données (chargement, stockage, sauvegarde) et plus globalement de tout leur fonctionnement.

Cette recherche de fiabilité commence évidemment par la conception et sa phase d'analyse. Les maquettes produites par PowerPoint sont très utiles, mais même si elles sont agrémentées de plus d'informations textuelles, elles ne portent pas toutes les problématiques techniques auxquelles il faudra répondre. Il est donc nécessaire de formaliser de manière plus détaillée les contraintes et problématiques pour qu'elles soient mieux identifiées et prises en compte dans la phase de développement.

Une autre des contraintes du projet était de ne pas être dépendant de technologies ou cadres qui imposeraient un niveau de connaissances spécifique et des contraintes techniques relatives à leurs évolutions. Cette logique est juste mais j'ai pu constater que l'évolution technologique est inévitable, que ce soit en PHP ou en jQuery.

D'autre part, le choix peut être fait de sélectionner un cadre léger [CIURLIK14] ou de s'en inspirer pour développer une architecture modulaire et maîtrisée et bénéficier de leurs services de bases : routage, réécriture d'URL, mise en cache ou sécurité des fichiers.

Il convient de rappeler que des cadres même récents sont le résultat de multiples évolutions et utilisent des bonnes pratiques éprouvées d'architecture logicielle. Il est donc difficile de concurrencer leurs performances et leurs capacités.

Le développement en mode objet est aussi une solution pour améliorer la modularité et l'organisation du code, ce qui entraîne un gain de fiabilité quand des évolutions de l'application sont apportées.

Ces types d'évolutions sont en rupture avec le contexte actuel, car la complexité va s'accroître. La mise en place de guides et d'une documentation plus importante peut cependant faciliter la prise en main. L'architecture structurée sur un modèle classique comme MVC doit faciliter la maîtrise globale de l'application. Les ingénieurs développement ont en effet l'habitude de rencontrer un tel concept.

Parmi les autres modifications qui pourraient être apportées, il pourrait être effectué une retouche du modèle de données afin de corriger la confusion qu'apporte la gestion simultanée des groupes et des rapports. Cela serait l'occasion de revoir le nommage de tables et champs pour en faciliter la gestion et ne pas utiliser de mots réservés.

C. Le bilan personnel

Depuis ma formation initiale de DUT informatique effectuée en année spéciale (post premier cycle) et au long de mon expérience professionnelle effectuée dans l'informatique système, je souhaitais pouvoir travailler sur la conception d'application métier. Ce stage m'a permis d'expérimenter des problématiques de développeur. Le cursus que j'ai suivi ces dernières années au CNAM n'était pas orienté « développement », ce qui m'a placé dans une situation où je devais trouver des solutions dans un environnement riche en informations. J'ai répondu aux problématiques de différentes manières. Pour certaines, j'ai proposé des solutions efficaces, comme des choix graphiques ou la gestion des téléchargements de fichiers. Pour d'autres, comme

la gestion des styles CSS, ce sont d'abord des questionnements qui se sont accumulés, et une évolution de mes pratiques qui s'est faite dans le temps à mesure que j'accumulais de la connaissance et me l'appropriais. Dans ce cas, je suis conscient que de la pratique est encore nécessaire pour arriver à un rendu et une méthode maîtrisée. D'autres éléments comme la conception ne peuvent pas être remis facilement en cause, car ils sont ponctuels et ne se réitèrent pas continuellement. Pour ce type de tâches, c'est avec une réflexion à posteriori qui peut permettre une amélioration des pratiques. Dans le cas de la conception, en parallèle de la création des maquettes avec PowerPoint, j'ai essayé de formaliser les spécifications de ce qui m'était demandé (cf. Annexes K). J'aurais pu utiliser l'UML, mais je le maîtrisais peu et celui-ci est orienté développement objet. Je pense à présent que j'aurais dû commencer par recueillir les besoins techniques et fonctionnels par des documents en langage naturel. Le travail aurait été ensuite d'organiser ces informations et il était possible d'aller plus loin dans la modélisation, au besoin, à partir de ce matériau.

Un autre sujet expérimenté, est l'importance de l'expertise. Sur différents domaines il n'a pas été facile, voire impossible, de valoriser des solutions car je ne les dominais pas totalement. Cette situation ne permet pas d'étayer pleinement la solution et n'est bien sûr pas rassurante pour l'interlocuteur. Je pense que si je devais faire à nouveau le stage avec mon expérience acquise, j'aborderais certaines problématiques différemment, avec bien plus d'assurance, et j'aurais des arguments étayés pour conforter mes propositions.

Cette expérience a été intense et m'a demandé beaucoup d'efforts, mais elle m'a permis de connaître le plaisir que l'on éprouve à concevoir des outils informatiques pour des utilisateurs. J'ai appris de nouvelles techniques, acquis des méthodologies, et ma curiosité n'en est que plus aiguisée face à l'ampleur des connaissances et expériences qui me restent à découvrir.

LISTE DES FIGURES

Figure 1 : Cas d'utilisation de LabBook pour un étudiant.	12
Figure 2 : Interface de choix des missions de l'étudiant.	13
Figure 3 : Interface d'édition de la mission étudiante.	14
Figure 4 : Interface du service de ressources.	15
Figure 5 : Interface du service de messagerie.	16
Figure 6 : Menu de gestion des LabDocs.....	16
Figure 7 : Outil LabDoc de texte TinyMCE.	17
Figure 8 : Outil de LabDoc dessin Drawex.....	17
Figure 9 : Outil de LabDoc COPEX, pour la création de protocoles expérimentaux.....	18
Figure 10 : Outil de LabDoc Fitex pour le traitement de données expérimentales.	19
Figure 11 : Cas d'utilisation de LabBook par un enseignant.	20
Figure 12 : Menu et boutons spécifiques aux enseignants (ancienne interface).	21
Figure 13 : Interfaces de gestion des classes (ancienne interface).	22
Figure 14 : Interface de gestion d'un groupe (ancienne interface).	23
Figure 15 : Exemple de maquette de la page de gestion des classes.	24
Figure 16 : Bandeau supérieur des pages dédiées aux enseignants (nouvelle interface).	25
Figure 17 : Ergonomie des entêtes de blocs et tableaux (nouvelle interface).	25
Figure 18 : Types d'icônes selon les systèmes d'exploitation, de gauche à droite : Linux, Apple et Microsoft.....	26
Figure 19 : Page de gestion des classes (nouvelle interface).	26
Figure 20 : Fenêtre d'importation d'étudiant par fichier au format csv (nouvelle interface).	27
Figure 21 : Page d'attribution des missions (nouvelle interface).	28
Figure 22 : Page de gestion des missions (nouvelle interface).	29
Figure 23 : Page de suivi des rapports (nouvelle interface).	30
Figure 24 : Première partie de la page d'édition des missions, dédiée aux paramètres de la mission.	31
Figure 25 : Deuxième partie de la fenêtre d'édition des missions, dédiée aux ressources.....	32
Figure 26 : Troisième partie de la page d'édition des missions, dédiée au rapport.....	32
Figure 27 : Ajout d'une partie de rapport dans la fenêtre d'édition des missions.....	33
Figure 28 : Ajout d'un LabDoc de l'enseignant dans la fenêtre d'édition des missions.	34
Figure 29 : Édition d'un LabDoc de l'enseignant dans la fenêtre d'édition des missions.....	34
Figure 30: Dossiers à la racine de l'application LabBook.....	37
Figure 31: Structure initiale des pages de l'application LabBook.....	38
Figure 32: Implémentation du modèle MVC dans LabBook.....	40
Figure 33 : Diagramme EER de la base de données LabBook, hors tables liées à la messagerie et aux commentaires	42
Figure 34: Exemple de blocs BEM [source bem.info].....	49
Figure 35 : Exemple d'éléments BEM [source bem.info].....	49
Figure 36: Architecture d'une page de l'interface des enseignants.	54
Figure 37: maquette sous PowerPoint 2013.....	60
Figure 38: Outil d'audit du navigateur Google Chrome.....	62
Figure 39 : Découpage modulaire de LabBook.	65
Figure 40: Exemple d'écran, de la maquette de l'outil COPEX	70
Figure 41: Commentaire de fonction	71

Figure 42: Information contextuelle sur une fonction, depuis un commentaire normalisé.....	72
Figure 43: Interface d'accueil générée par le logiciel Doxygen	72
Figure 44: Information sur une fonction dans le site généré par Doxygen	73

LISTE DES TABLEAUX

Tableau 1 : Liste des fonctions d'accès à la base de données.....	43
Tableau 2 : Champs de la base de données provenant d'entrées de l'utilisateur	44
Tableau 3 : Liste des fonctions contenues dans le fichier gestio_classes.js.....	47

LISTE DES EXTRAITS DE CODE

Extrait de code 1 : Exemple de requête PDO.....	43
Extrait de code 2 : Mise en variable du nom de la page web	46
Extrait de code 3 : Règles communes de style pour les pages de gestion enseignantes	51
Extrait de code 4 : Préparation des données au format JSON pour les variables JavaScript.....	56
Extrait de code 5 : Création de l'objet JavaScript à partir des données PHP au format JSON	56

BIBLIOGRAPHIE

- [Grafikart15] BOYER Jonathan. Grafikart [en ligne]. Disponible sur : <http://www.grafikart.fr/> (consulté en 09 2015).
- [Notepad++15] HO Don. Notepad++ [en ligne]. Disponible sur : <https://notepad-plus-plus.org/> (consulté en 09 2015).
- [GNU15] Free Software Foundation. GNU [en ligne]. Disponible sur : <http://www.gnu.org/copyleft/gpl.html> (consulté en 09 2015).
- [Sublime Text15] Sublime HQ Pty Ltd. Sublime Text [en ligne]. Disponible sur : <http://www.sublimetext.com/> (consulté en 09 2015).
- [SEVILLEJA10] SEVILLEJA Chris. (2010, Dec.) Best of Sublime Text 3: Features, Plugins, and Settings. [Online]. <https://scotch.io/bar-talk/best-of-sublime-text-3-features-plugins-and-settings>
- [Package Control15] BOND Will. Package Control [en ligne]. Disponible sur : <https://packagecontrol.io/> (consulté en 09 2015).
- [The Open Group15] Open Group. The Open Group [en ligne]. Disponible sur : <http://www.opengroup.org/subjectareas/enterprise/archimate> (consulté en 09 2015).
- [ISO03] ISO. (2003, June) ISO 10006:2003. Document. [Online]. http://www.iso.org/iso/fr/catalogue_detail?csnumber=36643
- [GestionDeProjet15] C.CORNIC. GestionDeProjet [en ligne]. Disponible sur : http://www.gestiondeprojet.net/articles/perimetre_projet.html (consulté en 09 2015).
- [agilealliance15] Agile Alliance. agilealliance [en ligne]. Disponible sur : <http://agilealliance.org/> (consulté en 09 2015).
- [Doxygen15] HEESCH Dimitri van. Doxygen [en ligne]. Disponible sur : <http://www.stack.nl/~dimitri/doxygen/index.html> (consulté en 10 2015).
- [OSMANI12] OSMANI Addy. (2012, July) Learning JavaScript Design Patterns. papier, électronique. [Online]. <http://addyosmani.com/resources/essentialjsdesignpatterns/book/#modulepatternjavascript>
- [ROSSOLINI08] ROSSOLINI Guillaume. (2008, July) PDO. [Online]. <http://php.developpez.com/faq/?page=pdo>

- [CSS15] SUTTERLITY Laurent. CSS [en ligne]. Disponible sur : <http://css.sutterlity.fr/les-bases-css/poids-selecteurs-css> (consulté en 10 2015).
- [stubbornella.org 15] SULLIVAN Nicole. stubbornella.org [en ligne]. Disponible sur : <http://www.stubbornella.org/content/> (consulté en 10 2015).
- [BEM15] YANDEX. BEM [en ligne]. Disponible sur : <https://en.bem.info/method/naming-convention/> (consulté en 10 2015).
- [SMACSS15] SNOOK Jonathan. SMACSS [en ligne]. Disponible sur : <https://smacss.com/pages/fr> (consulté en 10 2015).
- [Google Developers15] Google. Google Developers [en ligne]. Disponible sur : <https://developers.google.com/web/fundamentals/?hl=en> (consulté en 10 2015).
- [CIURLIK14] CIURLIK Yohann. Top 5 des micro-frameworks PHP du moment [en ligne]. , Jan. 2014 Disponible sur : <http://www.spawnrider.net/2014/01/13/top-5-des-micro-frameworks-php-du-moment/> (consulté en 10 2015).
- [tinymce15] Ephox Corporation. tinymce [en ligne]. Disponible sur : <http://www.tinymce.com/> (consulté en 10 2015).
- [Zwibbler15] Steve Hanov. Zwibbler [en ligne]. Disponible sur : <http://zwibbler.com/> (consulté en 10 2015).
- [E. CARRÉ14] E. CARRÉ M. PILVAR , "Evolution of a collaborative learning environment for experimental sciences," UFR SHS, Grenoble, Projet M1 WIC 2014.

ANNEXES

D. Diagramme de Gantt.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A																			
			B		B	B	B	B	B	B	B								
	C1	C2			C2	C3	C3	C4	C5										
																	D1	D1	D1

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
		B											B						
C6																			
D2	D2	D2	D2			D2	D2			D2	D3	D4	D4	D5	D5	D5		D5	
							E			E									
							F					F	F	F	F	F		F	

A - Prise en main de LabBook (*travail sur les documents du projet et l'UML*).

B – Auto formation.

C - Développement du module "Création des missions & pré-structuration du rapport"

« *Module de création des missions et de pré-structuration des rapports : ce module doit permettre aux enseignants de décrire une mission, c'est-à-dire, créer les textes d'introduction à la mission, la structure du rapport attendu, les ressources accessibles ainsi que les LabDocs fournis comme modèles.* »

C1 - Création des maquettes.

C2 – Développement des interfaces d'édition des missions.

C3 – Test et correctifs dans l'environnement de développement commun.

C4 – Tests utilisateurs dans l'environnement de développement.

C5 – Intégration des modifications demandées et correction des anomalies.

C6 – Intégration dans l'environnement de production.

D - Améliorer le module "Suivi enseignant"

« *Module de suivi du travail des étudiants : ce module est implémenté de façon non satisfaisante. Il sera nécessaire de rationaliser son code en refactorisant certaines fonctions communes avec d'autres parties du code.* »

D1 – Création des maquettes.

D2 – Développement des interfaces de l'enseignant.

D3 – Test et correctifs dans l'environnement de développement commun.

D4 – Tests utilisateurs dans l'environnement de développement.

D5 – Intégration des modifications demandées et correction des anomalies.

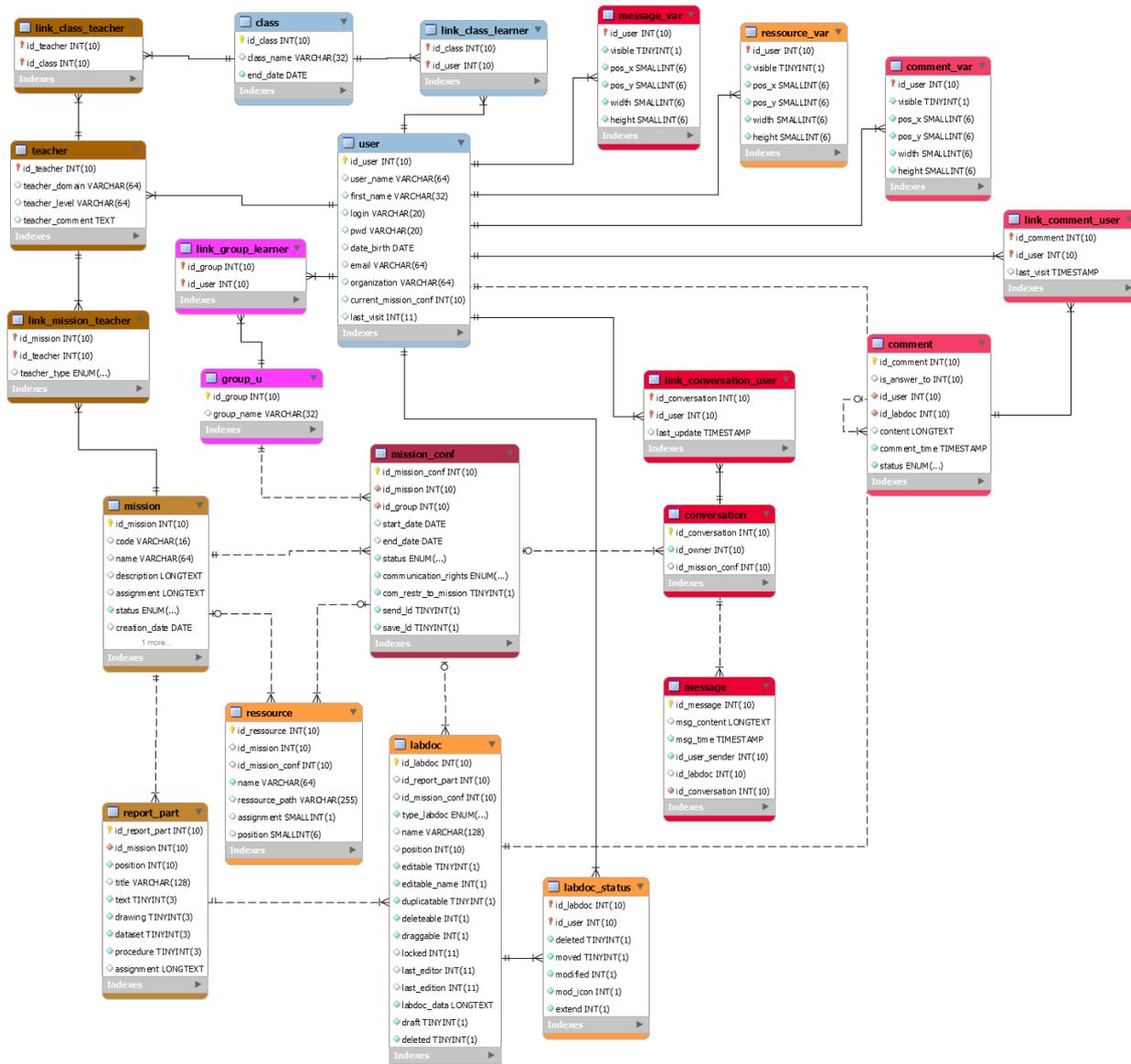
E - Améliorer la stabilité de l'exécution de l'outil Copex et étudier la mise à jour de son composant JSTree.

F – Élaboration du mémoire.

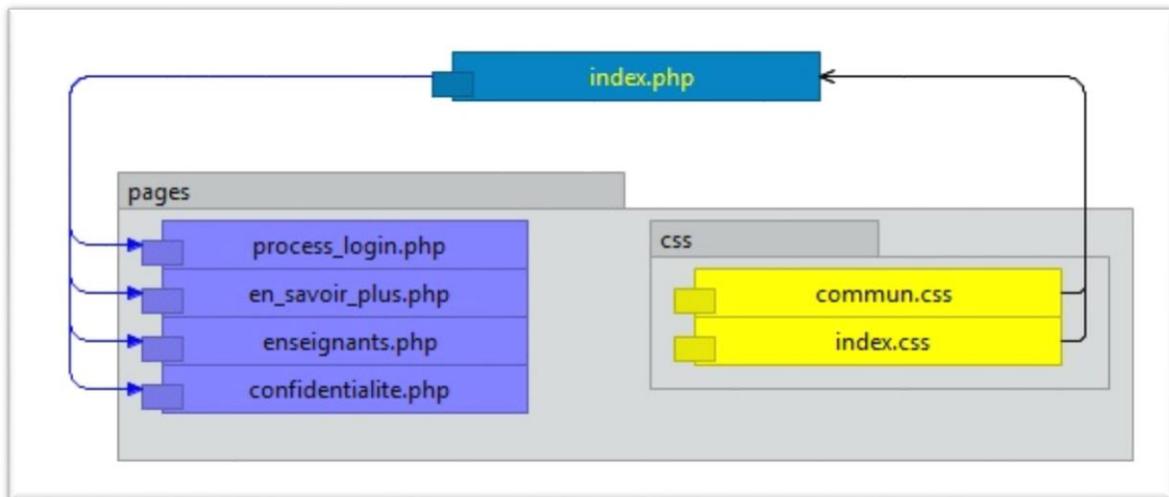
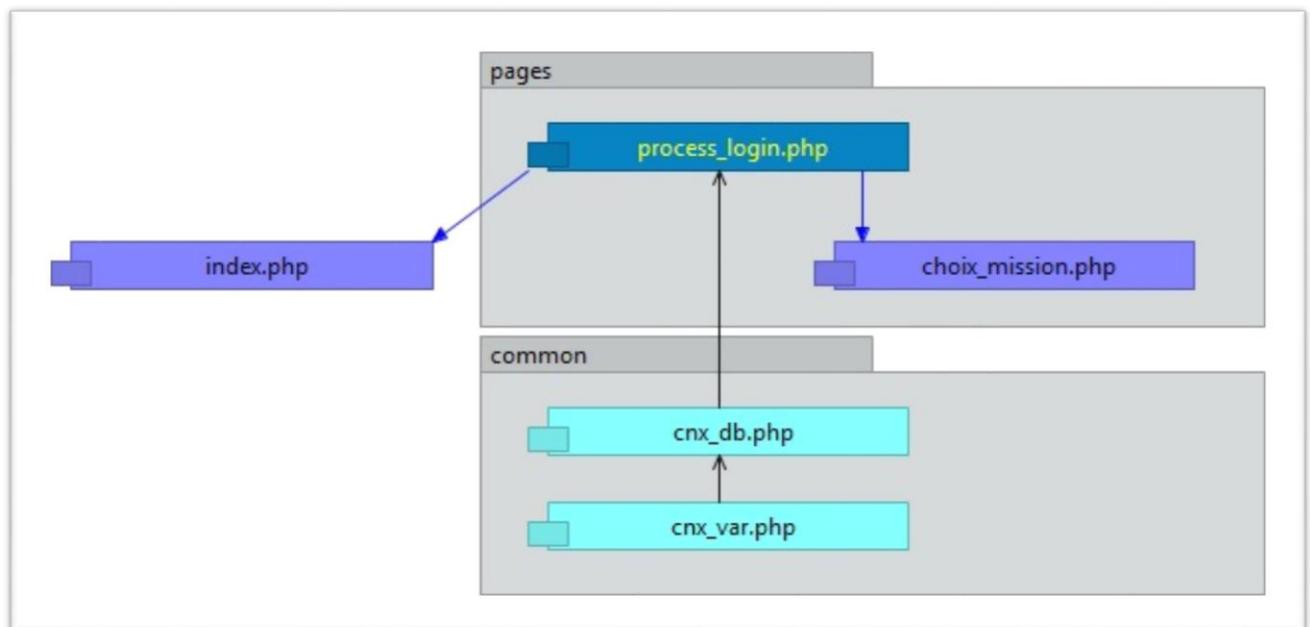
E. Diagramme de la base de données

1. Diagramme EER (Enhanced Entity Relationship)

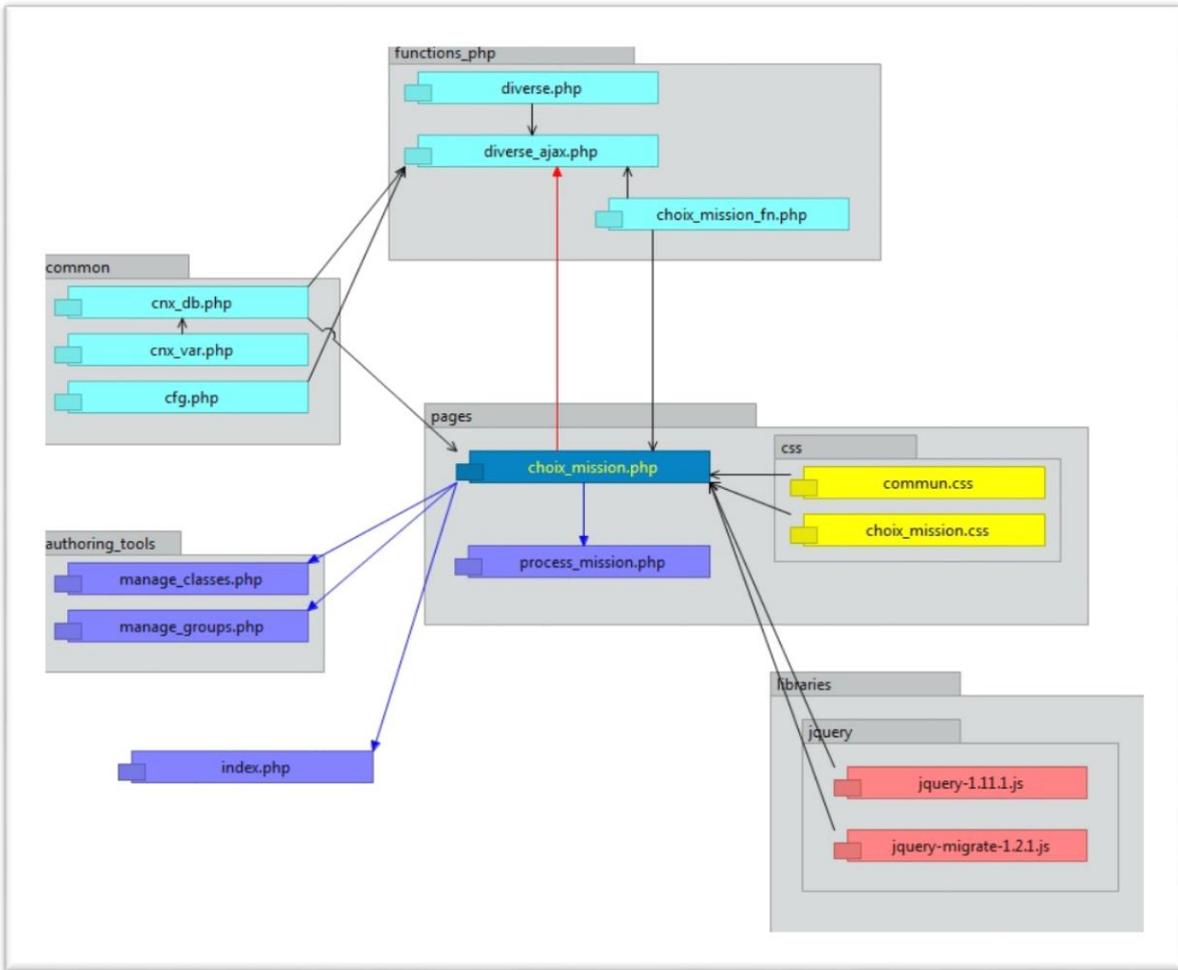
Diagramme généré par l'application MySQL Workbench des différentes tables de LabBook.



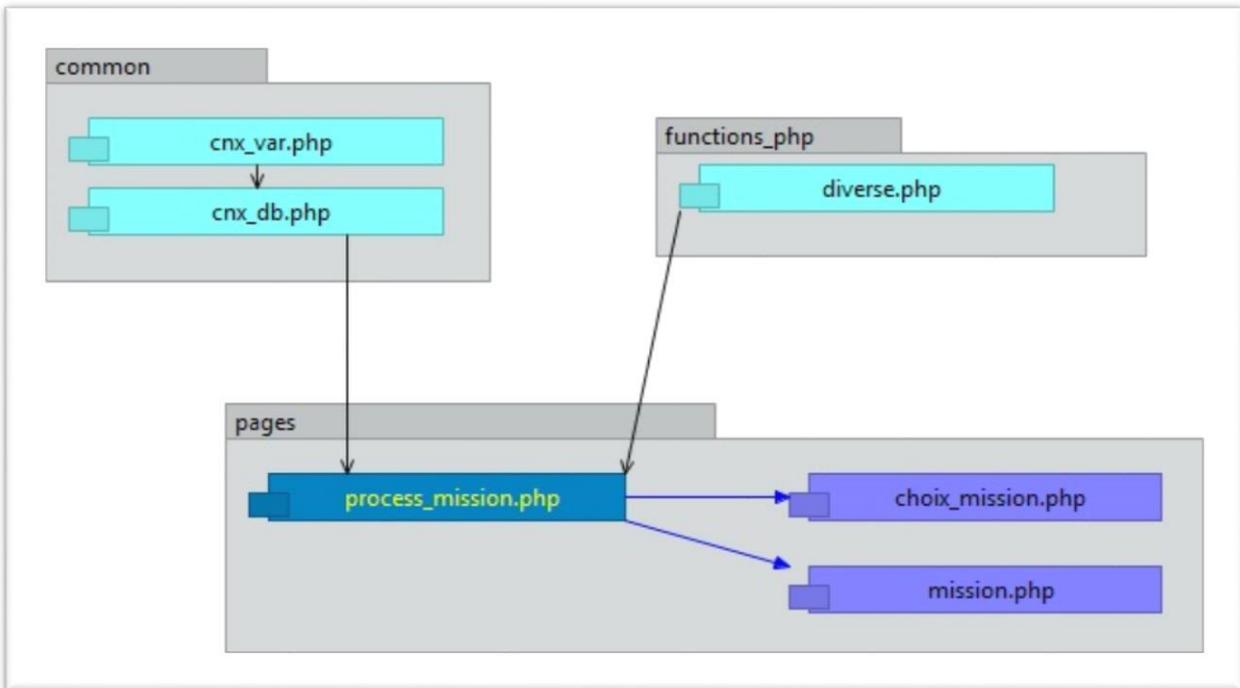
F. Schémas de la structure de fichiers des pages web.

1. Schéma de la page « *index.php* »2. Schéma de la page « *process_login.php* »

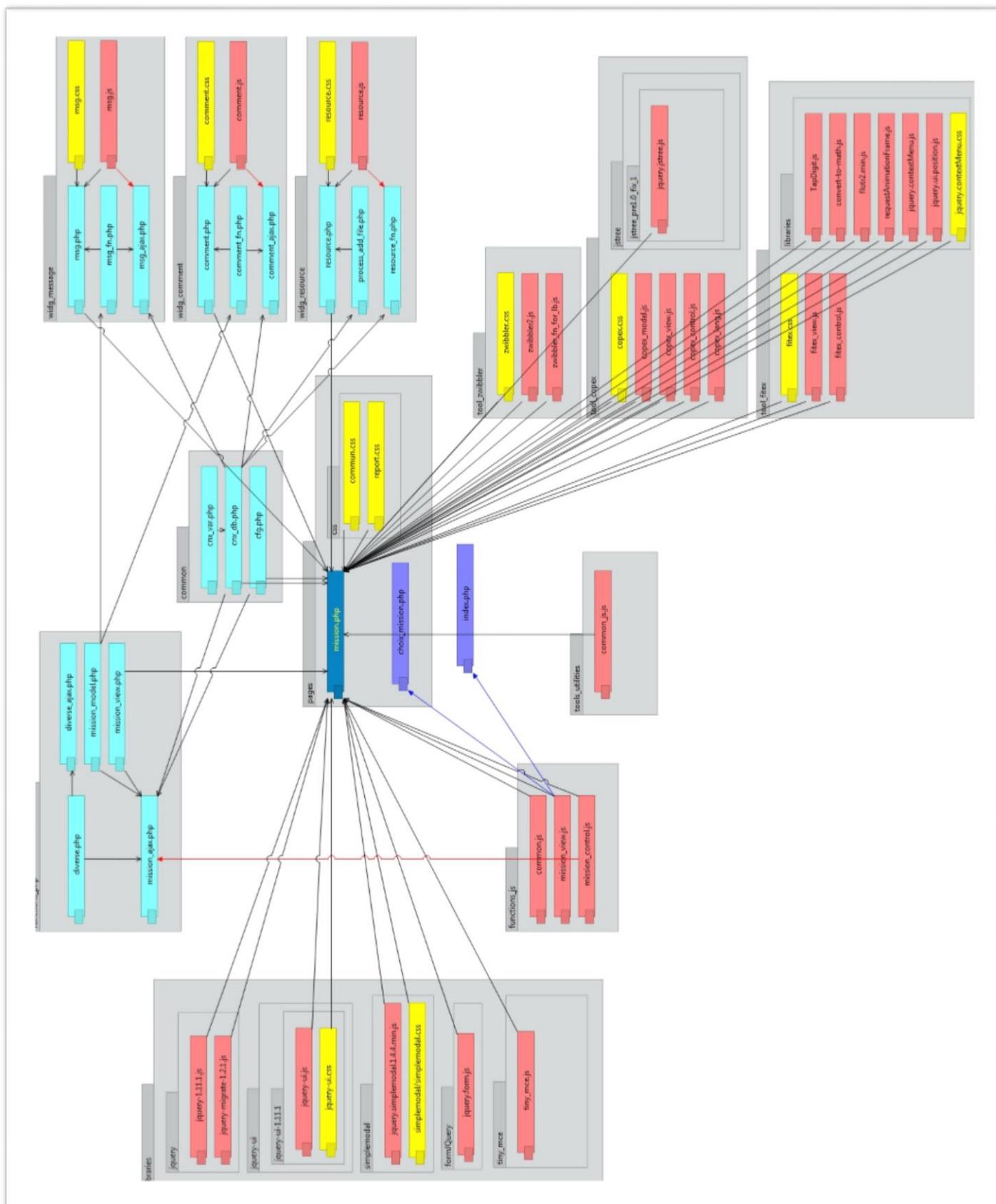
3. Schéma de la page « *choix_mission.php* »



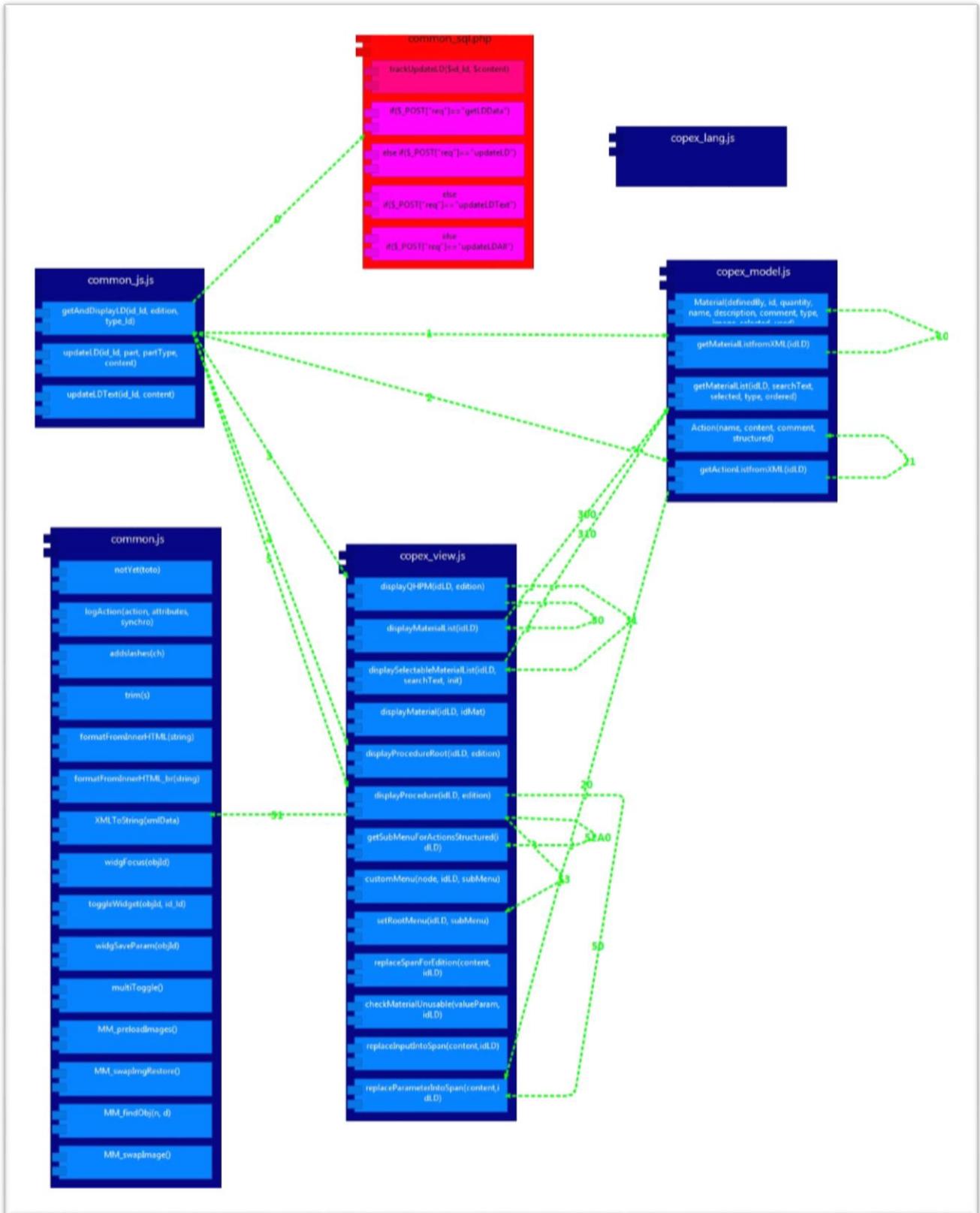
4. Schéma de la page « *process_mission.php* »



5. Schéma de la page « mission.php »



G. Schéma de la séquence des appels entre fonctions initiée par *getAndDisplayLD()*



I. Document (partiel) d'enregistrement des tests

PROJET : LABBOOK plateforme pédagogique spécifique aux enseignements scientifiques expérimentaux.	
Equipe : MeTAH, Modèles et Technologies pour l'Apprentissage Humain Responsable : Cédric D'HAM Stagiaire : Arnaud LE MEILLOUR	Date de création : 08/09/2015 Dernière modification : 08/09/2015
Page :	

I. Page : *gestion_classes.php*

labbook-dev1.ext/authoring_tools/gestion_classes.php

LabBook GESTION DES CLASSES Cédric d'Ham

CLASSES		ÉTUDIANTS			INFORMATIONS
Nom	Validité	NOM	Prénom	Conx.	étudiant
ABCDEFGHIJKLMNOP...	12-12-2016	AUCAIGNE	fanny	12-08-2015	Nom Buyse
DUT 2014	01-02-2016	BOURGUIGNON	jeremie		Prénom Pierre-Emmanuel
magistere_13	29-11-2013	BRONNEC	guillaume		Date de naissance / /
MASTER 2016	06-08-2017	BUYSE	pierre-emmanuel		E-mail
MyClass	01-12-2016	CAFOURNET	elodie		Organisation
Teachers	31-12-2099	DETRAIT	maximin		Login buysep
testalm4	06-06-2006	DUFOURD	thibault		Mot de passe *****
<input type="checkbox"/> Ne pas afficher les classes périmées		DUPOUY	henri		
		JACQUI	audrey		
		ZAMORANO	natalia		

La description des actions et tests se présente sous la forme d'un entête qui précise le contexte suivit de la description :

[OS/Browser][Zone de l'élément][Élément sur lequel l'action est faite][Action]

Description...

A. Zone liste des classes :

Affichage des 16 premiers caractères des noms de classes, suivi de trois points.

[Windows7/Chrome][Entête « CLASSES »][Icône d'ajout « + »][Clic gauche]

- Au chargement de la page, le bloc d'étudiant est vide et il n'y a pas de bloc d'options affiché. Le clic sur ajout fait apparaître le bloc d'ajout de classe et enlève la sélection si elle existe sur une classe donnée.
- Si la liste d'étudiants est affichée, ainsi qu'un autre bloc d'options, ces éléments sont cachés et toute sélection sur une classe ou un étudiant est supprimée.

CLASSES +	
Nom ▲	Validité
ABCDEFGHIJKLMNOP...	12-12-2016
DUT 2014	01-02-2016
magistere_13	29-11-2013
MASTER 2016	06-08-2017
MyClass	01-12-2016
Teachers	31-12-2099
testalm4	06-06-2006

Ne pas afficher les classes périmées

[Windows7/Chrome][Entête colonne][Case du nom][Clic gauche]

- Au chargement initial le tri est croissant sur cette colonne.
- A chaque clic, les tris bascule de croissant à décroissant et l'icône du triangle indique le sens du tri, et les données de la colonne sont mis à jour.

[Windows7/Chrome][Entête colonne][Case de validité][Clic gauche]

- Au clic sur l'entête de validité, un tri initial croissant s'effectue et le triangle précisant le sens du tri disparaît de la colonne où il était actif, et apparaît à droite du texte de l'entête.
- Comme pour tous les tris, le clic répété bascule le tri de croissant à décroissant.

[Windows7/Chrome][Colonne des noms de classe][Case d'un nom][Clic gauche]

- Au clic sur une case de nom de classe, la liste des étudiants apparaît dans le deuxième bloc, s'il y a des étudiants, ainsi que le descriptif de la classe dans le troisième bloc information.

[Windows7/Chrome][Colonne des validités de classe][Case d'une date][Clic gauche]

- Effet identique au clic sur la case du nom.

[Windows7/Chrome][Sous le tableau][Case à cocher de filtre sur la date][Clic gauche]

- Lorsque l'on coche la case du filtre des « classes périmées », la liste se met à jour tout en gardant le tri actif.
- Si une classe sélectionnée disparaît du fait de l'activation du filtre, la liste des étudiants apparaît, et les détails de la classe disparaissent.

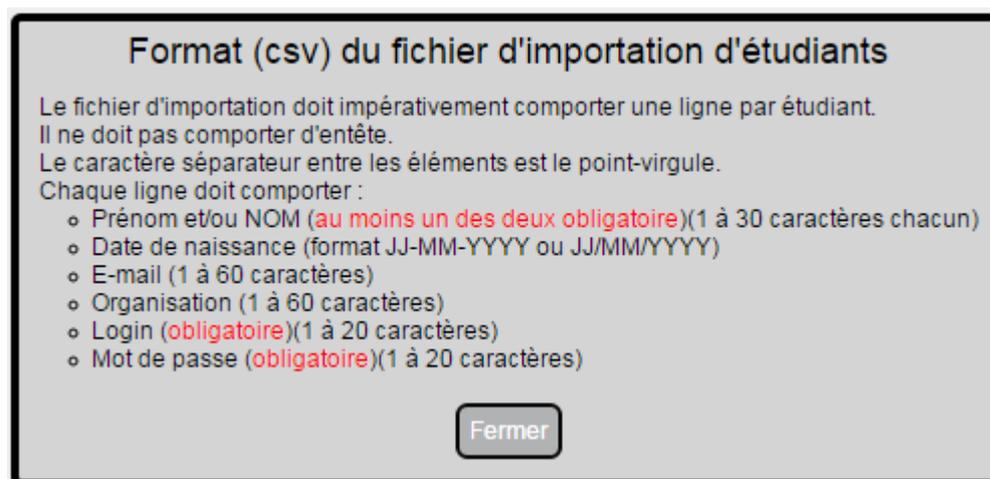
B. Zone de liste des étudiants :

ÉTUDIANTS		
NOM ▲	Prénom	Conx.
AUCAIGNE	fanny	12-08-2015
BOURGUIGNON	jeremie	
BRONNEC	guillaume	
BUYSE	pierre-emmanuel	
CAFOURNET	elodie	
DETRAIT	maximin	
DUFOURD	thibault	
DUPOUY	henri	
JACQUI	audrey	
ZAMORANO	natalia	

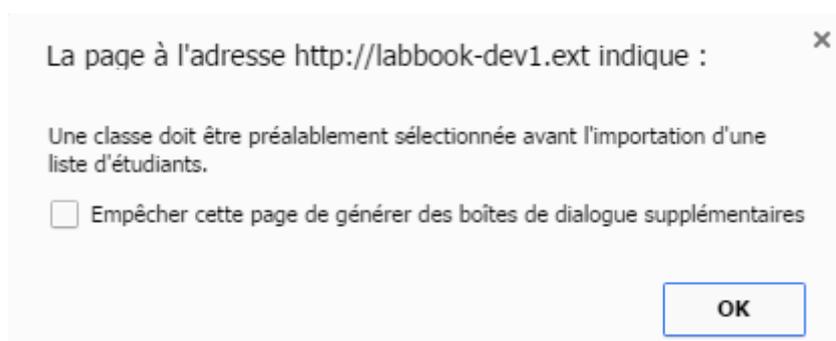
[Windows7/Chrome][Entête

« ÉTUDIANTS »][Icône d'information «  »][Clic gauche]

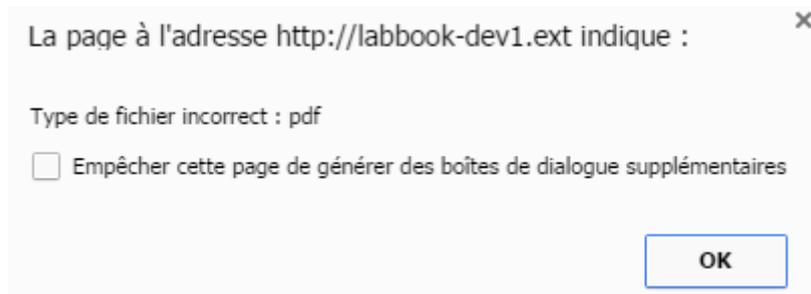
- Affichage d'une fenêtre d'information sur les spécifications du fichier .csv demandées pour l'importation d'étudiants.

[Windows7/Chrome][Entête « ÉTUDIANTS »][Icône d'importation «  »][Clic gauche]

- Le clic sur l'icône entraîne l'affichage d'une fenêtre de sélection de fichier local si une classe est préalablement sélectionnée, sinon un message d'erreur.



- L'utilisateur doit alors sélectionner un fichier .csv. Si le fichier n'a pas la bonne extension un message d'alerte est envoyé.



- Une fois le fichier chargé, une fenêtre affiche les enregistrements détectés et propose, par des cases à cocher, de choisir les enregistrements d'étudiants que l'on souhaite ajouter dans la classe sélectionnée.
- L'utilisateur a la possibilité de modifier les données affichées avant de valider directement dans le tableau.

IMPORTATION							
Sélectionnez les lignes à importer à l'aide des cases à cocher. Vous pouvez modifier le contenu des champs avant de valider. *Champs obligatoires : Nom, Prénom, Login, Mot de passe.							
	NOM*	Prénom*	Date de naissance	E-mail	Organisation	Login*	Mot de passe*
<input type="checkbox"/>	nom	prénom	date de naissance (format DD-MM-YYYY)	E-mail	Organisation	login	password
<input checked="" type="checkbox"/>	Romain	jean	12/01/1981	thomas@free.fr		toto05	toto05
<input checked="" type="checkbox"/>	tarot	francis	03-21-1986	roberdu-srd@free.fr	ENS3	toto04	toto04
<input type="checkbox"/>	lambert	raymond	04/06/1984	rlambert@gmail.com	DUT	toto06	toto06
<input type="checkbox"/>	nanty	isabelle		inanty@yahoo.com	DUT	toto07	toto07
<input type="checkbox"/>	bourdon		12/01/1981		DUT	toto08	toto08
<input type="checkbox"/>	Martin-laval	Pierre-Francois	04/06/1984	roberdu-srd@free.fr		toto09	toto09
<input type="checkbox"/>	ducret	Arnaud	12/04/1982	aducuret@free.fr		toto10	
<input type="checkbox"/>	Celma	Stéfi	03-21-86			toto11	toto11
<input type="checkbox"/>	Hajdi	youssef	04/06/1984		BTS	toto12	toto12
<input type="checkbox"/>	blanc	Louis-emmanuel	12/01/1981	roberdu-srd@free.fr	BTS	toto13	toto13
<input type="checkbox"/>	Meurisse	nina	12/04/1982		BTS	toto14	toto14
<input type="checkbox"/>			12/01/1967			toto141	
<input type="checkbox"/>	salvador	thomas	04/06/1984	roberdu-srd@free.fr	BTS	toto15	toto15
<input type="checkbox"/>	iones	tobv				toto16	toto16

- Une fois validée, la fenêtre de gestion des classes est rechargée et revient sur la classe et l'affichage de la liste des étudiants, mise à jour.

[Windows7/Chrome][Entête « ÉTUDIANTS »][Icône d'ajout «  »][Clic gauche]

- Entraîne l'affichage du bloc d'ajout d'un étudiant et le retrait d'autres blocs d'options si affichés.

- Vérifie si une classe est préalablement sélectionnée, sinon affiche :



[Windows7/Chrome][Entête colonne][Case NOM/Prénom/Cnx][Clic gauche]

- Au clic sur une des cases, mise à jour de la liste et du tri.
- Ordre du tri initial croissant.
- Boucle de l'ordre du tri croissant/décroissant sur clic répété de l'entête.

[Windows7/Chrome][Liste des étudiants][Ligne d'un étudiant][Click gauche]

- Au clic sur une des lignes d'un étudiant, mise en surbrillance de la ligne et affichage dans le bloc d'options, des informations de l'étudiant.

C. Zone bloc d'options d'ajout d'une classe :

[Windows7/Chrome][Entête colonne][Icône d'annulation]  **][Clic gauche]**

- Supprime l'affichage du bloc d'ajout.

[Windows7/Chrome][Entête colonne][Icône de Validation]  **][Clic gauche]**

- Enregistrement de la nouvelle classe et rechargement de la page avec la nouvelle classe sélectionnée et le bloc d'information sur la classe affichée.
- Contrôle sur la taille du nom de la classe :



AJOUT ✓ ✕

Classe

Nom

Date de validité
31 / 08 / 20 16

Enseignants associés

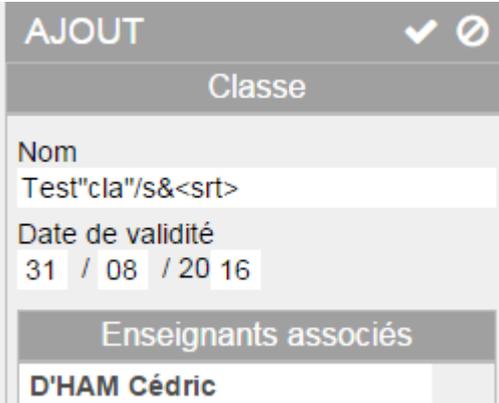
D'HAM Cédric

Enseignants non-associés

GIRAULT Isabelle	+
HÉRITIER Claudine	+
LE MEILLOUR Arnaud	+
MARZIN Patricia	+
TILQUIN François	+
WAJEMAN Claire	+

[Windows7/Chrome][Champ d'entrée][input Nom][input]

- Test d'entrée de valeur : vide
- Test d'entrée de valeur : > 30 char => blocage des entrées à 30 caractères (idem Firefox 40)
- Test d'entrée de valeur avec caractères spéciaux
- Suppression des guillemets lors de l'enregistrement.



AJOUT ✓ ✕

Classe

Nom
Test"cla"/s<srt>

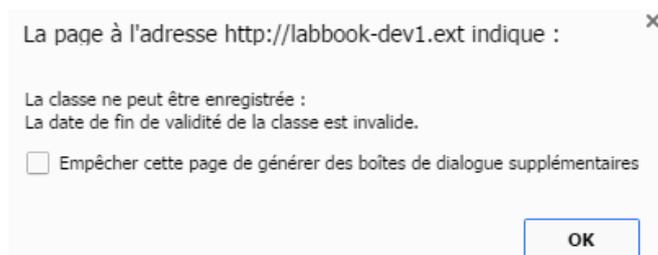
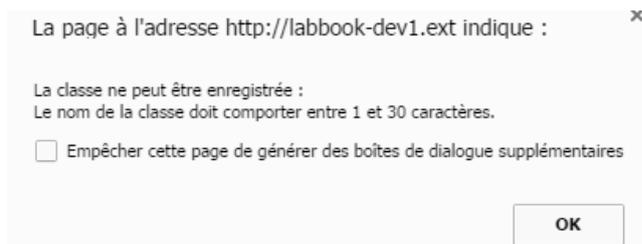
Date de validité
31 / 08 / 20 16

Enseignants associés

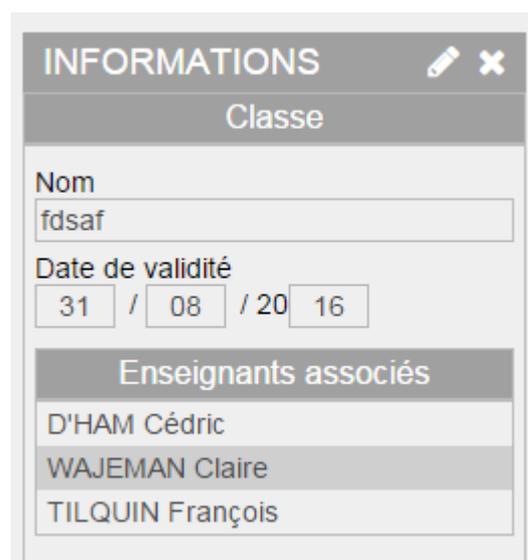
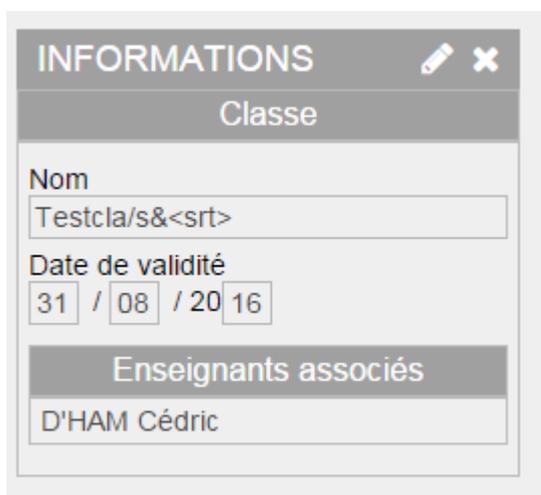
D'HAM Cédric

[Windows7/Chrome][Champ d'entrée][input Date][input]

- Contrôle de la validité de la date :



D. Zone bloc d'options d'information de la classe :

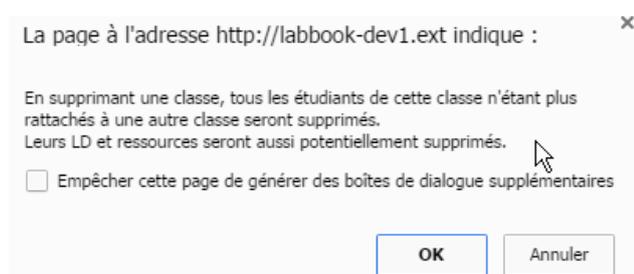


[Windows7/Chrome][Entête colonne][Icône d'édition

- Bascule en mode édition et cache la fenêtre d'information.

[Windows7/Chrome][Entête colonne][Icône de suppression

- Message d'alerte avant suppression de la classe



K. Document (partiel) de spécification de la page *edit_mission.php*

PROJET : LABBOOK plateforme pédagogique spécifique aux enseignements scientifiques expérimentaux.	
Equipe : MeTAH, Modèles et Technologies pour l'Apprentissage Humain Responsable : Cédric D'HAM Stagiaire : Arnaud LE MEILLOUR	Date de création : 16/03/2015 Dernière modification : 31/03/2015
Page : edit_mission.php	

I. Expression des besoins :

1. Exigences fonctionnelles

La page d'édition des missions doit permettre d'effectuer toutes les opérations nécessaires à la création et à la modification des paramètres d'une mission.

Seuls les enseignants ont accès à cette page, et doivent avoir le moins de contraintes possibles sur la manière dont ils souhaitent travailler leur mission. L'enregistrement des modifications se fera donc en permanence par des requêtes AJAX.

2. Exigences non-fonctionnelles

La prise en main de l'application par l'enseignant doit être rapide. Pour cela l'ergonomie doit être sobre et l'utilisation intuitive. Des aides contextuelles permettront à l'utilisateur d'obtenir les précisions nécessaires.

Les deux autres points essentiels doivent être la rapidité et la fiabilité de l'application.

Exigences générales sur le développement :

- Apparence :
 - o Adaptation des blocs d'éléments tant que les tailles minimales des éléments ne sont pas atteintes.
 - o Fonctionnement les navigateurs web :
 - ~~Ms internet explorer v xx et supérieur~~
 - Mozilla Firefox v xx et supérieur
 - Google chrome v xx et supérieur
 - ~~Apple safari v xx et supérieur~~
- Codage :
 - o Respecter au mieux la sémantique du contenu.
 - o Ne pas anticiper des besoins et contraintes non encore exprimés.
 - o Éviter la redondance.
 - o Commenter le code pour en faciliter la maintenance.
 - o Appel des fonctions JS, par les « events » HTML onXXX depuis le DOM.
 - o Limiter le nombre de requêtes AJAX au minimum.
 - o Ne paramétrer les requêtes AJAX en mode Synchrone qu'en cas d'obligation.
 - o Ne passer et tester les arguments des fonctions que si nécessaire au vue du contexte. Si la fonction fait partie de la page, le contexte est connu, si elle fait partie de fichiers « communs » le contexte est incertain.
 - o Veiller à ce que les valeurs de retour des fonctions soient utiles.

II. Spécification de la page globale :

La page est découpée en trois parties principales : paramètres, ressources et rapport.

- La page est appelée avec une requête `$_GET['` id_mission`'] = int`

L'absence d'information ``` id_mission``` au chargement de la page, ou une valeur du paramètre à `-1` déclenche la création d'un nouvel id dans la table mission et l'édition d'une nouvelle mission (fonction `createMission()`).

Si la variable `$_GET` existe et que la valeur de l'id de la mission est valide, les informations de la mission sont chargées et affichées dans la page pour une édition de mission existante.



The screenshot shows the 'PARAMETRES DE LA MISSION' section of the LabBook interface. At the top left is the LabBook logo, and at the top right is the user name 'Cédric d'Ham' with a refresh icon. The main content area is titled 'PARAMETRES DE LA MISSION' and contains several form elements:

- Code:** A text input field containing 'Dragons'.
- Nom:** A text input field containing 'Etude préalable pour le projet Dragons durables'.
- Enseignant(s) associé(s):** A table with three rows, each representing an instructor. The first row has a '+ Ajouter un enseignant' button. The second row shows 'Isabelle Girault' as 'Concepteur'. The third row shows 'Arnaud LE MEILLOUR' as 'Tuteur'. The fourth row shows 'Patricia Marzin' as 'Tuteur'. Each row has a dropdown menu for the role and a delete 'X' icon.
- Statut:** A dropdown menu currently set to 'Création'.
- Texte de présentation:** A text area containing the following text: 'Cette mission est une étude scientifique préalable pour un projet artisanal en développement durable. Dans le projet *Dragons durables*, on envisage d'utiliser des dragons pour produire de la chaleur avec un impact environnemental quasi nul (modulo les artisans croqués). Les dragons finlandais sont employés par les souffleurs de verre (température de flamme comprise entre 600°C et

Si la variable `$_GET` est absente ou n'est pas un entier, que l'id donné est négatif ou ne correspond pas à l'id d'une mission existante, la page de création de mission s'initie avec des champs vides. Seuls les champs Code et Nom sont accessibles.

- Contrôle du statut d'enseignant au chargement, par l'id_user, rejet si non valide.

LabBook Cédric d'Ham

PARAMETRES DE LA MISSION

L'édition de la mission nécessite la validation du code (2 à 12 caractères), et du nom (4 à 64 caractères), de la mission.

Code

Nom

Enseignant(s) associé(s) ⓘ Statut ⓘ

+ Ajouter un enseignant Création ▼

Texte de présentation

RESSOURCES

Consigne courte

III. Descriptions des éléments de la partie paramètre :

1. Bloc « code » et « Nom » :

A – Initialisation :

Insertion de la valeur depuis la table mission si valide, sinon champ vide.

B – Traitement dynamique :

Contrôle sur la valeur du champ code qui ne doit pas correspondre à un code déjà présent dans la base.

Contrôle de la longueur des chaînes entrées à la sortie du champ.

- Si les deux champs sont valides, création de la mission et retrait du calque afin de poursuivre l'édition de la mission.
- Si l'entrée est invalide, un message d'information est envoyé et le champ reprend la valeur précédente.

C – Commentaire :

Stockage de la valeur initiale dans un SPAN caché.

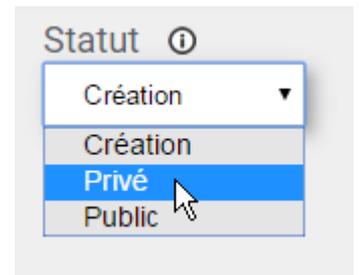
2. Bloc « statut » :

A – Initialisation :

Si une *mission_conf* liée à la mission courante existe, l'option « Création » de la liste Statut est désactivée.

B – Traitement dynamique :

Le champ correspondant est mis à jour dans la BD.



3. Bloc « enseignement associé » :

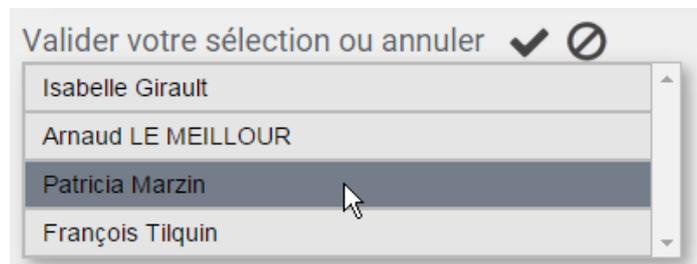


A – Initialisation :

Affichage tabulaire des enseignants associés avec pour chacun une liste déroulante permettant de basculer du statut de Tuteur à Concepteur et inversement. Une icône propose la dissociation de l'enseignant à la mission à la fin de chaque ligne.

B – Traitement dynamique :

Au clic sur le bouton d'ajout des enseignants non-associés, une liste de sélection multiple apparaît. La touche Ctrl permet de sélectionner plusieurs enseignants et deux icônes permettent d'annuler l'opération ou d'effectuer la requête.



4. Bloc « texte de présentation » et « consigne courte » :

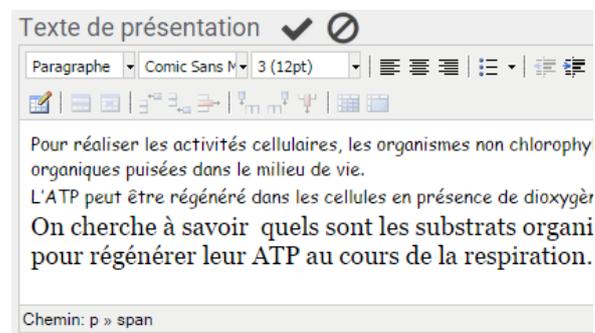
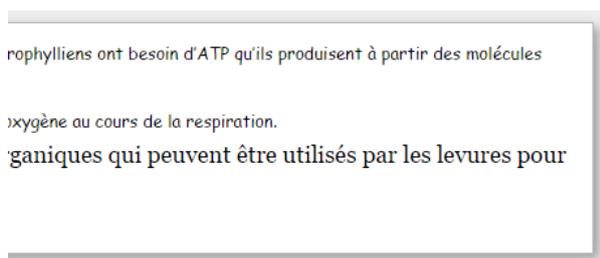
A – Initialisation :

Reprend les données de la BD si existantes.

B – Traitement dynamique :

Au clic sur le cadre, le bloc d'édition TinyMCE s'affiche pour l'édition du texte.

L'utilisateur a le choix de valider ou annuler sa saisie.



Mise en œuvre des évolutions de LabBook, plateforme collaborative d'apprentissage pour les sciences expérimentales.

Arnaud LE MEILLOUR

Grenoble, le 25/01/2016

Résumé

L'équipe MeTAH du Laboratoire Informatique de Grenoble travaille sur plusieurs projets de recherche dans le domaine des systèmes interactifs et cognitifs. LabBook, l'un de ses projets initié en 2010, a pour objectif l'impact sur l'apprentissage, de l'étayage de l'activité de conception expérimentale par structuration de la tâche et production de rétroactions individualisées.

Ce type d'application support à la recherche, est en constante maintenance pour répondre aux requêtes d'évolutions fonctionnelles. Elles nécessitent aussi des interventions dues aux contraintes technologiques ou pour corriger des anomalies.

C'est dans ce contexte que j'ai mené mon projet de refonte de l'interface dédiée aux enseignants. Après avoir analysé l'existant et les demandes fonctionnelles, j'ai conçu et réalisé l'ensemble de cette architecture. Ce travail a été l'occasion d'une réflexion sur les techniques utilisées et leurs homogénéités dans l'application.

Mots clés : EIAH, rapport scientifique, ergonomie

Summary

The team MeTAH Grenoble Informatics Laboratory is working on several research projects in the field of interactive and cognitive systems. LabBook is one of its projects initiated in 2010, which aims to have impact on the learning, on the propping up of the activity of experimental design by structuring the task and the production of individualized feedback.

This type of application support of research is in constant maintenance to meet the functional development requests. They also require interventions due to technological constraints or to correct anomalies.

It is in this context that I conducted my proposed redesign of the interface dedicated to teachers. After analyzing the existing and functional requirements, I designed and built this entire architecture. This work was an opportunity to reflect on the techniques used and their homogeneity in the application.

Keywords : ITS, scientific report, ergonomics