



**INFO**

Département Informatique  
IUT Belfort-Montbéliard

# Application de gestion des heures d'enseignement

## RAPPORT DE PROJET DE FIN D'ÉTUDE

*Du lundi 4 mai 2020 au jeudi 11 juin 2020*

*Enseignant référent: **Stéphane Domas***

*Étudiant: **Antoine RICHARD***

**2<sup>ème</sup>** année de DUT Informatique





## Sommaire

Introduction.....	4
1. Cahier des charges.....	5
1.1. Objectif du projet.....	5
1.2. Fonctionnalités.....	5
2. Mise en oeuvre.....	6
2.1. Choix des outils.....	6
2.2. Réalisation.....	7
2.2.1. Partie enseignant.....	7
2.2.2. Partie décomposition.....	8
2.2.3 Partie projet.....	10
2.2.4. Charte graphique.....	13
2.3. Difficultés rencontrées et solutions apportées.....	14
3. Bilan.....	15
3.1. Bilan pédagogique.....	15
3.2. Améliorations possibles.....	15
Conclusion.....	16
Table des illustrations.....	17

# Introduction

Dans le cadre de la deuxième année de DUT Informatique, nous devions effectuer un stage conventionné en entreprise d'une durée de 10 semaines. Hélas, en raison des conditions sanitaires et de l'interdiction de déplacement instauré en France du 17 mars 2020 au 11 mai 2020, l'entreprise dans laquelle j'aurai dû faire mon stage n'a pas pu m'accueillir. En conséquence de cela, la direction de l'IUT a pris des mesures en offrant la possibilité aux étudiants dans l'impossibilité de réaliser le stage d'effectuer un projet de fin d'étude du 4 mai au 12 juin. Ce projet est décomposé en deux parties. La première partie consistait en l'élaboration d'une liste collective de sujets. Ensuite nous avons pu sélectionner parmi cette liste cinq sujets que nous voulions faire. Enfin, un sujet nous a été attribué parmi les cinq choix. Plusieurs étudiants travaillent sur chaque sujet non pas en collaboration mais en compétition, cela permet une mise en concurrence et donc pousse chaque étudiant à faire son maximum pour avoir un meilleur rendu que ces concurrents.

Le sujet qui m'a été attribué est la réalisation d'une application web de gestion des heures d'enseignement. Un sujet proposé par Stéphane DOMAS, maître de conférences, et qui est de ce fait mon enseignant référent pour ce projet. Mon choix s'est porté sur ce sujet car il représente un défi assez intéressant sur le plan technique. Je me suis également dit que c'est un bon projet pour apprendre de nouvelles technologies.

# 1. Cahier des charges

## 1.1. Objectif du projet

Le projet de gestion des heures d'enseignement a pour but de permettre, via une application web, de faire un bilan des heures faites par chaque enseignant afin de voir si ce dernier respecte le nombre d'heures obligatoires et supplémentaires selon son statut. Dans le but d'effectuer ce bilan, l'utilisateur de l'application pourra créer la décomposition de la formation qu'il souhaite, puis pour chaque matière il pourra définir son type de saisie des heures entre hebdomadaire et global. Puis remplir les heures de chaque formation dans un tableau qui se doit d'être simple d'utilisation et ergonomique.

## 1.2. Fonctionnalités

Les fonctionnalités devant être présentes dans l'application sont nombreuses. Premièrement, il y a des données qui seront intemporelles et donc qui ne changeront pas au fil des années. C'est le cas des enseignants, il faut pouvoir en ajouter, modifier, supprimer, dupliquer. Il faut également leur attribuer un statut avec des heures minimal et maximal attendu et supplémentaires. L'application doit avoir la possibilité de représenter la structure d'une formation (décomposition) en arbre. Ensuite chaque année il faut pouvoir créer des projets et leur associer des décompositions pour créer des formations. Il faut ensuite pouvoir ajouter aux projets des intervenants tirés de la liste des enseignants. Chaque formation aura des éléments de niveau 1 représentant une période et ayant un nombre de semaines. Ensuite ces périodes auront des sous-éléments avec un type de saisie (aucun, hebdomadaire, global). Pour les éléments ayant aucun type de saisie ils pourront avoir des éléments enfants. Mais pas les autres types, pour ces derniers il faudra choisir le type de cours autorisé (TP, TD, CM, PARTIEL). Le type de saisie hebdomadaire indique que pour chaque type de cours autorisé et pour chaque semaine de la période dans laquelle se trouve l'élément (la matière) il faut indiquer le nombre d'heures effectué par groupes d'étudiants dans cette matière et ensuite le nombre de groupes dont s'occupe chaque intervenant de la matière. Le type de saisie globale indique que l'utilisateur doit définir des forfaits pour chaque type de cours autorisé et ensuite un volume par intervenant et par type de cours.

## 2. Mise en oeuvre

### 2.1. Choix des outils

L'application est construite avec **Express.js**, un framework Nodejs permettant de réaliser des applications web rapidement. Les principaux avantages d'Express sont premièrement le système de routage qui permet de spécifier des chemins utilisables par l'utilisateur et de définir avec quelles méthodes HTTP (GET, POST, PUT, ...) ces chemins sont accessibles. Ensuite Express permet de transmettre des variables aux moteurs de template (**Ejs** pour ma part) qui génèrent la page. J'ai choisi Ejs car ce moteur permet de combiner simplement de l'HTML et du JavaScript. Cependant pour les pages avec des rendus plus complexes ou demandant beaucoup d'interactions avec la base de données (page de saisie des heures, page des bilans, ...) j'utilise du **JavaScript pur** avec l'api **Fetch** qui permet d'effectuer des requêtes HTTP. Enfin la validation des données des requêtes se fait avec le module **express-validator**.

Concernant la base de données, étant donné que les décompositions des formations sont représentées sous la forme d'un arbre (éléments avec des sous-éléments) dans les maquettes officielles et dû au fait que de nombreuses données doivent être dupliquées d'année en année, j'ai opté pour un système de gestion de base de donnée noSQL orienté document : **MongoBD**. En effet cette solution permet d'imbriquer des données (appelées documents) ce qui facilite la gestion de la structure en arbre des formations et donc simplifie les requêtes et l'affichage de cette structure. J'utilise le module **Mongoose** de Nodejs comme passerelle entre l'application et la base de données. Mongoose utilise des « Schema » pour modéliser les données. Ces Schema permettent de définir les types de variables et de structurer les données. Grâce à ces Schema il est possible de créer des modèles de données. J'ai ajouté à cela le plug-in **mongoose-data-tree** qui va gérer la structure en arbre et va offrir des fonctions utiles pour pouvoir afficher cet arbre. La justification de l'utilisation de ce plug-in sera démontrée plus tard, dans la partie 2.2.

Enfin, au niveau du style j'utilise le framework CSS **Bulma** car il est entièrement responsive et personnalisable en utilisant le préprocesseur CSS **SASS**. Cela permet de pouvoir redéfinir certaines variables de Bulma.

Pour conclure, ces choix ont été faits dans le but de me simplifier la tâche: le développement côté serveur et côté client s'effectue dans le même langage (JavaScript) ce qui permet de gagner du temps. De plus MongoDB organise ses données en JSON, un format de données du langage JavaScript. Tous ces choix m'ont permis d'organiser mon code selon l'architecture modèle-vue-contrôleur. Les modèles étant générés par Mongoose, les vues par Ejs et les contrôleurs sont écrits avec Express.

## 2.2. Réalisation

L'application web se découpe en 3 parties: la partie enseignant, la partie décomposition et enfin la partie projet. Ces 3 parties sont facilement repérables dans l'application car elles forment le menu de navigation.

### 2.2.1. Partie enseignant
















Liste des enseignants					
Ajouter un enseignant		Gérer les statuts			
Nom		Recherche...			
Nom	Prénom	Surnom	Email	Statut	Actions
Dupond	Nicolas	NDU	nico@dpt.yopmail	Vacataire	  
Dupond	Phillibert	PDU	phil@dpt.yopmail	Professeur Certifié	  
Dupont	Jean	JDU	jean-dupont@number.fr	Maître de Conférence	  
Thierry	Jean-Michel	JMT	thierry@in.txt	Heures obligatoires: 192 - 192 Heures supplémentaires: 0 - 100	  
Michel	Hervé	HMI	herve@test.fr	Professeur des Universités	  

Illustration 1- Tableau des enseignants

La partie enseignant est la partie la plus basique de l'application. Elle permet simplement de créer et de gérer les enseignants et leurs status. Pour ajouter un enseignant il suffit de cliquer sur le bouton «Ajouter un enseignant» et un formulaire apparaîtra afin de compléter les données sur l'enseignant et notamment choisir son statut. Une fois ajouté, l'enseignant se retrouve dans le tableau (cf: Illustration 1). Chaque ligne du tableau représente un enseignant avec ses données. Si nous survolons avec la souris le statut de l'enseignant, un petit encart avec les descriptifs

des heures du statut apparaît. Il y a ensuite la possibilité d'effectuer des actions avec chacune de ces lignes : la modification, la duplication (qui ne crée pas directement un nouvel enseignant mais ouvre le formulaire d'ajout avec les champs déjà remplis) et la suppression (avec une validation si l'enseignant intervient dans un ou plusieurs projets). Dans le cahier des charges il était spécifié de pouvoir filtrer les lignes du tableau selon certains attributs, c'est pour cela qu'au-dessus du tableau il y a un champ de saisie et un sélecteur qui permettent de chercher dans le tableau une entité en fonction de l'attribut sélectionné (cf: Illustration 2).

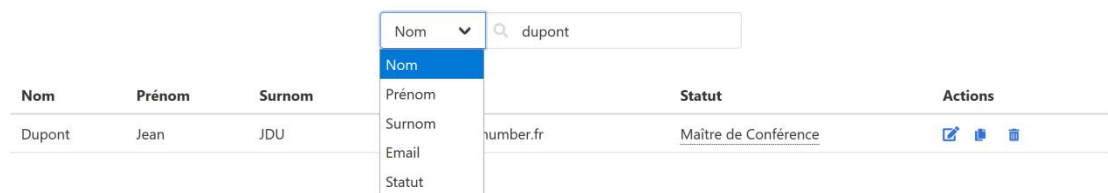


Illustration 2- Système de recherche dans le tableau

Concernant les statuts, il faut cliquer sur le bouton «Gérer les statuts» pour pouvoir y accéder. La page des statuts est en soi là même que celle des enseignants avec seulement des données différentes. De plus le bouton de suppression d'un statut ne s'affiche que s'il n'a aucun enseignant associé. Quand on crée un statut il faut lui associer un algorithme de calcul des heures.

Les pages de gestion des enseignants et des statuts sont entièrement générées avec Ejs car elles sont simples dans leurs constructions (avec une boucle *for* basique).

### 2.2.2. Partie décomposition

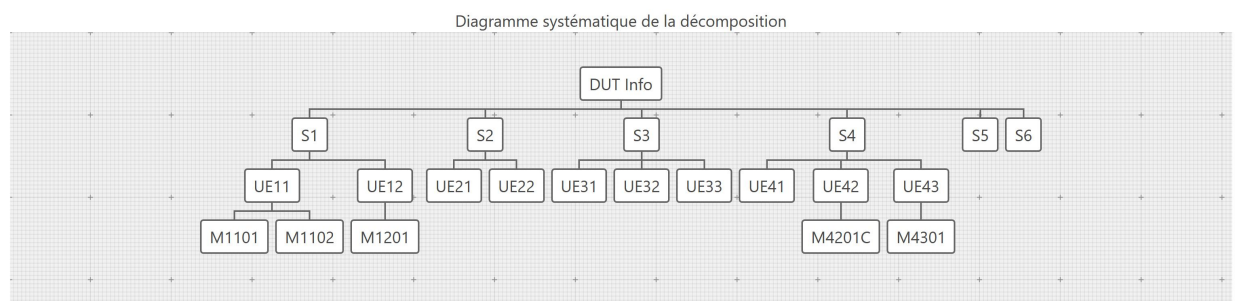


Illustration 3- Affichage d'une décomposition



Cette partie a été nettement plus complexe à mettre en place car chaque élément d'une décomposition peut avoir des éléments enfant. C'est pour cela que le j'utilise le plug-in mongoose-data-tree. En effet ce dernier permet de simplement spécifier le parent de l'élément que nous ajoutons dans la base de données. Cela crée deux attributs à cet élément. L'attribut «parent» qui est une référence sur l'identifiant unique de son parent et le champ «path» qui donne le chemin de l'élément en partant de l'élément racine de la décomposition. Ensuite le plug-in offre différentes fonctions dont «*getChildrenTree()*» qui permet d'obtenir les éléments de la décomposition directement imbriqués les uns dans les autres, donc chaque élément aura un tableau contenant ses enfants dans un attribut nommé «children». Grâce à cela l'affichage (cf: Illustration 3) et le traitement de la décomposition sous la forme d'un arbre est facilité car il suffit de faire une fonction récursive sur chaque enfant de chaque élément.

Quand on crée une décomposition l'élément racine est directement créé. Puis en cliquant sur l'élément une fenêtre modale (cf: Illustration 4) montrant les propriétés de l'élément s'ouvre et présente des boutons permettant de modifier l'élément ou d'ajouter un enfant à cet élément. La page est générée en JavaScript pur car il est plus facile de faire des fonctions récursives avec plutôt qu'avec Ejs.

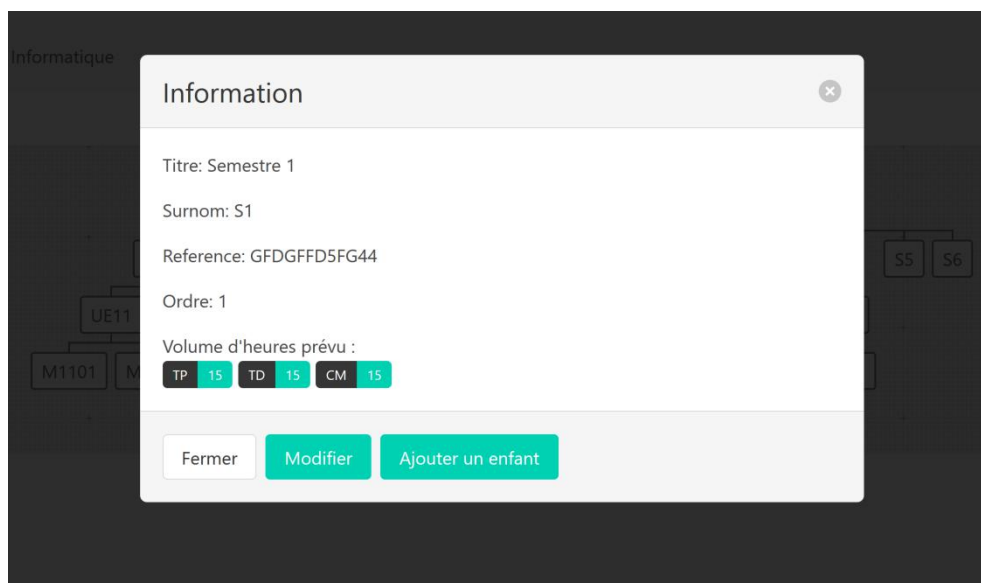


Illustration 4- Fenêtre modale de gestion d'un élément constitutif

### 2.2.3 Partie projet

La partie projet est la partie la plus complexe de l'application car elle contient trois autres parties: intervenant, formation et bilan. La partie projet en elle-même est juste représentée par une liste de projets (cf: Illustration 5) ayant chacun une liste de formations et d'un bouton permettant d'ajouter une formation. La création d'un projet peut se faire en copiant ou non un projet d'une année précédente. Une fois un projet créé il y a la possibilité de modifier ce projet, de voir les bilans des heures, de gérer les intervenants du projet ou encore d'archiver le projet. Le fait d'archiver un projet fait qu'il ne soit plus visible dans la page générale des projets mais dans la page des projets archivés. Il est ensuite possible de restaurer les projets archivés pour qu'ils apparaissent à nouveau dans la page générale.

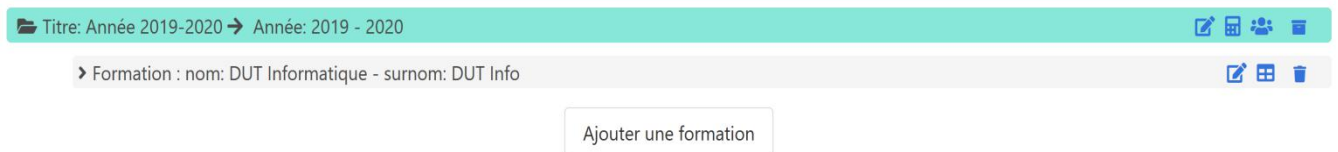


Illustration 5- Affichage d'un projet

#### 2.2.3.1. Intervenant

La partie de gestion des intervenants est générée avec du JavaScript pur et utilise l'API Fetch car j'ai voulu que quand on ajoute, retire ou modifie un intervenant la page ne se recharge pas et ainsi cela augmente la fluidité pour l'utilisateur. En effet si nous voulons ajouter des intervenants il faut cliquer sur le bouton «Ajouter des intervenants», alors une fenêtre modale (cf: Illustration 6) va s'ouvrir avec des champs de saisie des heures obligatoires et supplémentaires pour chaque enseignant, ces champs sont déjà préremplis avec les valeurs du statut des enseignants. Le fait d'utiliser Fetch permet que quand nous ajoutons un intervenant la fenêtre modale reste ouvert et se met à jour. Identique pour le tableau des intervenants si nous modifions ou retirons des intervenants le tableau va se mettre à jour sans recharger la page. De plus une notification de réussite ou d'échec va apparaître après chaque action effectuée. Enfin le même système de recherche que pour les enseignants est mis en place.

Liste des enseignants




Nom

Recherche...

Nom	Prénom	Surnom	H obligatoires		H supplémentaires		Statut	Action
Dupond	Nicolas	NDU	0	64	0	0	Vacataire	+
Dupont	Jean	JDU	192	192	0	100	Maître de Conférence	+ Ajouter
Michel	Hervé	HMI	192	192	0	100	Professeur des Universités	+ Fermer

Illustration 6- Fenêtre modale d'ajout d'intervenants dans un projet

### 2.2.3.2. Formation

Pour créer une formation il faut lui attribuer une décomposition créée précédemment, ainsi des éléments pédagogiques sont créés avec une référence sur les éléments constitutifs de la décomposition. La racine de cette décomposition devient donc le diplôme, les éléments de niveau 1 deviennent des périodes. Puis pour chaque élément n'ayant pas d'enfant il est possible de définir son type de saisie. Il y a également la possibilité d'ajouter des périodes ou des éléments dans la formation qui n'auront pas de référence sur un élément constitutif de la décomposition. Pour l'affichage des formations, au vu de sa complexité et dans un souci d'ergonomie, j'utilise également du JavaScript pur et l'API Fetch. Afficher une formation revient à afficher sa hiérarchie dans un tableau. Pour les éléments dont qui ont un type de saisie hebdomadaire ou global les cases de saisie dans le tableau sont directement modifiables en cliquant dessus. Pour ajouter des intervenants dans un de ces éléments il faut cliquer sur le bouton  ,  pour en retirer. Puis quand nous voulons sauvegarder les saisies entrées pour un élément il faut appuyer sur le bouton  de l'élément. Pour un maximum de lisibilité il est possible de plier et déplier les éléments enfants d'un élément. Enfin comme pour les décompositions, quand on clique sur un élément une fenêtre modale s'ouvre et affiche les informations sur l'élément.

DUT INFORMATIQUE						
Semestre 1						
		1	2	3	4	total
UE 11 : Bases de l'informatique						
M1101: Introduction aux systèmes informatiques						
Système	TP	1	1	1		3
	TD	1	1	1		3
	CM	1.5	1.5	1.5		4.5
	PARTIEL				2	2
Jean-Michel Thierry	TP	1	1	1		3
	TD	2	2	2		6
	CM	4	4	4		18
	PARTIEL				1	2
Phillibert Dupond	TP					0
	TD	1	1	1		3
	CM	2	2	2		9
	PARTIEL					0

Illustration 7- Type de saisie hebdomadaire

Saisies globales				
		stage	projet	total HeTD
Nicolas Dupond	TP			0
	TD	2	3	32
	CM			0
	PARTIEL			0
Phillibert Dupond	TP			0
	TD	4	1	24
	CM			0
	PARTIEL			0

Illustration 8- Saisies globales pour une période

### 2.2.3.3. Bilan

En cliquant sur le bouton bilan du projet, une page comme l'illustration 9 apparaît. Sur cette page, les boutons définissent les codes couleurs et en cliquant dessus seuls les enseignants ayant un HeTD respectant le nom du bouton vont apparaître. De plus une recherche par prénom, nom et statut est possible. Comme pour la page formation, les calculs du bilan et des heures ne sont pas stockés dans la base de données et sont donc effectués à chaque appel à la page. J'ai choisi de faire cela car il était plus facile pour moi de faire ce calcul quand on appelle la page plutôt

que de modifier les champs de la base de données à chaque changement dans les heures.

TousBonSous serviceLimite dépassée

PrénomRecherche...

Phillibert Dupond   Professeur Certifié   H oblig. : 384 - 384   H supp. : 0 - 200						
Type	TP	TD	CM	PARTIEL	H supplémentaire	HeTD
Global	60	28	9	0	0	101.5

Nicolas Dupond   Vacataire   H oblig. : 0 - 64   H supp. : 0 - 0						
Type	TP	TD	CM	PARTIEL	H supplémentaire	HeTD
Global	16	32	0	0	0	48

Hervé Michel   Professeur des Universités   H oblig. : 192 - 192   H supp. : 0 - 100						
Type	TP	TD	CM	PARTIEL	H supplémentaire	HeTD
Global	0	801	0	0	609	801

Illustration 9- Affichage du bilan d'un projet

2.2.4. Charte graphique

Chaque page de l’application est visuellement organisée selon le modèle si dessous :

Gestion des heures d'enseignementProjetsDécompositionsEnseignants

Titre de la page

Contenu de la page

Réalisation de Antoine RICHARD dans le cadre du projet de fin d'étude.

Illustration 10- Modèle de page

Seul le texte «Titre de la page» et la section «Contenu de la page» changent en fonction de la page à afficher, les autres éléments sont communs à toutes les pages . J'ai choisi cette structure de page car c'est la plus commune pour une application web. De ce fait chaque utilisateur sait intuitivement où trouver les informations qu'il cherche.

Concernant la palette de couleurs j'ai choisi des couleurs plutôt sobres et froides. J'ai opté pour du vert comme couleur principale car c'est une couleur qui exprime l'apaisement et qui n'agresse pas les yeux. Puis les boutons d'actions ont une couleur bleue car cette dernière permet d'attirer l'attention et de se démarquer des autres éléments. Le bleu dans un site web est également associé à l'interactivité notamment car la plupart du temps les liens hypertextes ont une couleur bleue. L'utilisateur saura donc qu'en appuyant sur un élément bleu il y aura de l'interactivité. Enfin le gris permet de mettre en valeur des éléments secondaires car il contraste avec le blanc du fond de la page mais est moins mis en valeur qu'un élément avec une couleur verte.



Illustration 11- Palette de couleur

### 2.3. Difficultés rencontrées et solutions apportées

Ma principale difficulté rencontrée est une difficulté concernant le délai. En effet, j'ai choisi des technologies avec lesquelles je n'ai jamais travaillé (Express, MongoDB, Ejs) mais qui sont plus adaptés pour ce projet, il m'a donc fallu un temps d'adaptation. Étant donné que le projet était assez long à faire et qu'il n'y a eu que six semaines pour l'effectuer j'ai donc dû travailler plus d'heure que ce qui était prévu durant ces six semaines. Mais je pense avoir réussi à surmonter cette difficulté notamment en regardant des cours et en consultant les documentations des technologies dès que j'avais des problèmes. De plus à la moindre hésitation quant à la compréhension d'une partie du cahier des charges je demandais directement des compléments d'informations à mon professeur référent ce qui m'a permis de ne pas rester bloqué sur un problème.

## 3. Bilan

### 3.1. Bilan pédagogique

Au niveau de la partie technique, j'ai appris énormément durant ce projet. Notamment concernant Express et MongoDB, deux technologies que je n'avais pratiquement jamais utilisées. En effet j'utilise généralement le langage PHP côté serveur et très rarement le JavaScript, j'ai donc dû m'adapter à des nouvelles règles et de nouveaux aspects du langage (promesses, fonctions asynchrones, ...). De plus j'ai pu renforcer mes compétences en SASS et en JavaScript Vanilla.

### 3.2. Améliorations possibles

Plusieurs améliorations peuvent être effectuées sur mon projet. Premièrement il y aurait la finalisation des tâches que je n'ai pas eu le temps de faire ou qui ont des potentiels bugs. Le design peut également être retravaillé car il n'était pas le sujet principal et il est donc assez simpliste.

Ensuite concernant le projet en général des fonctionnalités pourraient être ajoutées, notamment l'exportation et l'importation des données sous d'autres formats (pdf, csv, ...) afin que ces données puissent être utilisées dans d'autres applications (par exemple des tableurs) ou qu'elles soient plus lisibles et formelles. Le fait de pouvoir importer les données permettrait de remplir les saisies plus rapidement.

Enfin, une autre fonctionnalité pouvant être ajoutée est la création des algorithmes de calculs des heures par l'utilisateur. Cela permettrait une plus grande généralisation de l'application.

## Conclusion

J'estime que mon projet respecte la plupart des demandes du cahier des charges. Il reste cependant certains points que je n'ai pas eu le temps de traiter mais dans l'ensemble les fonctionnalités prioritaires ont été traitées.

Le fait de ne pas avoir pu effectuer un stage en entreprise est regrettable puisque j'aurais pu apprendre le travail dans une équipe expérimentée et surtout les normes en entreprise. Mais ce projet représente assez bien un projet pouvant être effectué en entreprise en raison de sa complexité et son cahier des charges très détaillé.

Je pense donc que ce projet a parfaitement complété ma formation puisque j'ai pu mettre en pratique mes connaissances apprises durant les deux années mais j'ai surtout pu améliorer ma capacité à m'adapter à des technologies inconnues. A défaut de travail en équipe, ce projet m'a permis d'apprendre à travailler seul et efficacement sur un projet plus complexe que ce que nous avions en cours. Le fait d'être en concurrence avec d'autres étudiants a fait que je m'investisse encore plus dans le projet pour essayer d'avoir un meilleur rendu que les autres. De plus la concurrence est un aspect très présent en entreprise, que ce soit entre employés ou entre sociétés, ce fût donc un bon entraînement.



## Table des illustrations

Illustration 1 - Tableau des enseignants.....	7
Illustration 2 - Système de recherche dans le tableau.....	8
Illustration 3 - Affichage d'une décomposition.....	8
Illustration 4 - Fenêtre modale de gestion d'un élément constitutif.....	9
Illustration 5 - Affichage d'un projet.....	10
Illustration 6 - Fenêtre modale d'ajout d'intervenants dans un projet.....	11
Illustration 7 - Type de saisie hebdomadaire.....	12
Illustration 8 - Saisies globales pour une période.....	12
Illustration 9 - Affichage du bilan d'un projet.....	13
Illustration 10 - Modèle de page.....	13
Illustration 11 - Palette de couleur.....	14