

**Rapport de projet :**

**Création d’un site Web de gestion des cursus des étudiants**

**UE LO07**

**Semestre P17**

**CAO Cheng – ISI3**

**LEBRAUD Delphine – ISI2**

Table des matières

[Introduction 3](#_Toc486268731)

[I. Analyse du cahier des charges et des fonctionnalités 4](#_Toc486268732)

[A) Liste des contraintes 4](#_Toc486268733)

[B) Fonctionnalités demandées pour le site : 4](#_Toc486268734)

[C) Modélisation des fonctionnalités 5](#_Toc486268735)

[II. Etapes de conception du projet 5](#_Toc486268736)

[A) Organisation et répartition des tâches 5](#_Toc486268737)

[B) Technologies utilisées 5](#_Toc486268738)

[C) Difficultés rencontrées 6](#_Toc486268739)

[III. Etapes de conception de la base de données 7](#_Toc486268740)

[A) Schéma de la base de données 7](#_Toc486268741)

[B) Modèle physique des données 7](#_Toc486268742)

[C) Explications de la base de données 8](#_Toc486268743)

[Conclusion 9](#_Toc486268744)

# Introduction

Ce rapport présente les étapes de conception et de réalisation du projet de LO07 « Technologies du Web » réalisé au printemps 2017.

L’objectif de ce projet est de réaliser un site Web dynamique permettant une facilitation de la gestion des cursus des étudiants de l’UTT en ISI. Les étudiants doivent obtenir 180 crédits et respecter les règles du règlement des études pour être diplômés.

Or ces règles sont parfois compliquées et mal connues des étudiants, qui ne comprennent alors pas bien leur propre cursus et ne parviennent pas à le gérer. Le site devra alors simplifier la compréhension de ces règlements et la gestion de leur parcours.

Nous détaillerons et justifierons dans ce rapport les décisions que nous avons eu à prendre lors de la phase de conception ainsi que les choix techniques, afin de respecter au mieux es spécifications techniques et les fonctionnalités présentées dans le cahier des charges.

Nous présenterons dans une première partie les fonctionnalités spécifiées dans le cahier des charges. Puis nous détaillerons et expliquerons les étapes de conception du projet ainsi que celles de la base de données.

# Analyse du cahier des charges et des fonctionnalités

## Liste des contraintes

Le projet sera réalisé en utilisant exclusivement l'une des suites Web suivantes WAMP / LAMP /MAMP.  
Le site devra utiliser les éléments suivants :

* Un serveur Apache
* Des pages HTML
* Des feuilles de styles CSS pour toutes les pages de votre site
* Des scripts PHP
* Des instructions Javascript
* Éventuellement des éléments de Bootstrap, JQuery, etc ...
* Une base de données relationnelle pour la persistance des données
* Un Framework est optionnel

## Fonctionnalités demandées pour le site :

*Fonctionnalité 1 : collecte des informations sur des cursus via des formulaires*

Les étudiants peuvent décrire leur cursus via des formulaires Web dynamiques. Les informations de ces formulaires seront mémorisées dans une base de données.

Chaque élément de formation ajouté doit être relié à un cursus par un identifiant unique mais également à un étudiant par son numéro d’étudiant. Les formulaires doivent être dynamiques c’est-à-dire qu’ils doivent pouvoir renseigner autant d’éléments de formation que l’utilisateur le souhaite.   
La base de données devra relier des éléments de formation, des cursus et des étudiants.

*Fonctionnalité 2 : visualisation des informations des cursus*

Votre projet doit permettre de présenter un cursus dans une page Web. Vous présenterez les informations (les éléments d’un cursus), des agrégats (SUM, EXIST) permettant de connaître rapidement l’avancement dans le cursus et un ensemble d’indicateurs (SE validé, stages effectués, …).

Il faut donc une page récapitulative du cursus, qui collecte les données de la base de données. Elle doit représenter les éléments de formation de façon catégorisée selon le semestre, le TC ou la filière, et le type d’élément. La présentation la plus claire est probablement un tableau présentant des sous-totaux et des totaux de crédits.

*Fonctionnalité 3 : conformité des cursus via la description de règlements*

Le site doit pouvoir valider le cursus en fonction du règlement des études et afficher les éléments manquants s’il n’est pas conforme. Il prendra en compte le règlement actuel et le futur règlement proposé dans le sujet et l’étudiant pourra choisir à quel règlement il veut comparer son cursus.

*Fonctionnalité 4 : Export et import de cursus via des fichiers csv*   
Le site doit pouvoir importer un cursus sous la forme d’un fichier CSV et exporter au format CSV les cursus de la base de données.

## C:\Users\LENOVO\AppData\Local\Microsoft\Windows\INetCache\Content.Word\UseCaseDiagram1.jpgModélisation des fonctionnalités

# Etapes de conception du projet

## Organisation et répartition des tâches

Nous avons analysé les fonctionnalités puis réalisé une maquette papier des pages prévues pour le site. Ensuite nous avons créé la base de données ensemble.

Ensuite la répartition s’est faite comme suit :

Etudiant 1 :  
Développement « back-end » ( gestion des formulaires php et connexion à la base de données, gestion des règlements et affichage des cursus)

Etudiant 2 :  
Modélisation des fonctionnalités, développement « front-end » (création des formulaires dynamiques, design du site avec CSS et Bootstrap, architecture et ergonomie)  
Pour des raisons évidentes de maîtrise du Français, un seul étudiant a rédigé le rapport afin d’optimiser le temps de rédaction (l’autre étudiant étant issu d’une université étrangère, le Français n’est pas sa langue maternelle).

***Github***

## Architecture du site

***Faire diagramme architecture***

## Technologies utilisées

Nous avons développé le site avec des pages html et réalisé la mise en page avec le framework Bootstrap. La gestion des formulaires a été faite avec le PHP, lié au SQL***. (netbeans)*** La base de données a été administrée avec le logiciel MySQLWorkbench ***(plus efficace).*** On a utilisé Amazon Web Service pour héberger notre site, afin de créer un véritable site web. Nous avons créé une instance EC2.

Fonctionnalité 1 : collecte des informations sur des cursus via des formulaires  
Pour les formulaires de description de cursus et d’étudiant, nous avons utilisé une méthode « GET » pour transférer les variables. Le formulaire form\_cursus.html a été dynamisé par une fonction Javascript « ajouteLigne() » qui permet de dupliquer les champs pour ajouter plusieurs éléments à un même cursus.

Fonctionnalité 2 : visualisation des informations des cursus  
Pour faire le lien entre PHP et MySQL (connexion, sélection), nous avons utilisé la fonction "mysqli".  
La fonction "fetch\_assoc()" a été utilisée pour récupérer chaque ligne de requête et les transférer dans un tableau. Pour créer des tableaux dynamiques d’affichage des cursus, nous avons utilisé une boucle « While ».

Fonctionnalité 3 : conformité des cursus via la description de règlements  
Nous avons utilisé la fonction "document.getElementById("\*").innerHTML" dans le JavaScript pour récupérer les éléments dans les tableaux d'affichage. Ensuite la fonction "parseInt" transfère les données pour faire les calculs. Enfin la fonction "window.alert()" affiche les messages décrivant le statut du cursus par rapport aux règlements et signale les éléments manquants.

Fonctionnalité 4 : Export et import de cursus via des fichiers CSV  
Pour l’import, nous avons utilisé des formulaires de type "enctype="multipart/form-data" pour uploader les fichiers, ainsi que la superglobale $\_FILES[\*.csv] pour récupérer le fichier. Afin de lire le contenu du fichier, nous avons utilisé la fonction "file\_get\_contents".

Pour l’export, fonction "fputcsv" écrit les données dans un fichier CSV. Nous avons utilisé un"header('Content-Disposition: attachment; filename=NOM\_Prenom.csv;')" pour permettre au navigateur de télécharger le fichier CSV automatiquement. Nous avons aussi utilisé une onction "fpassthru" pour afficher le reste des données dans le fichier CSV.

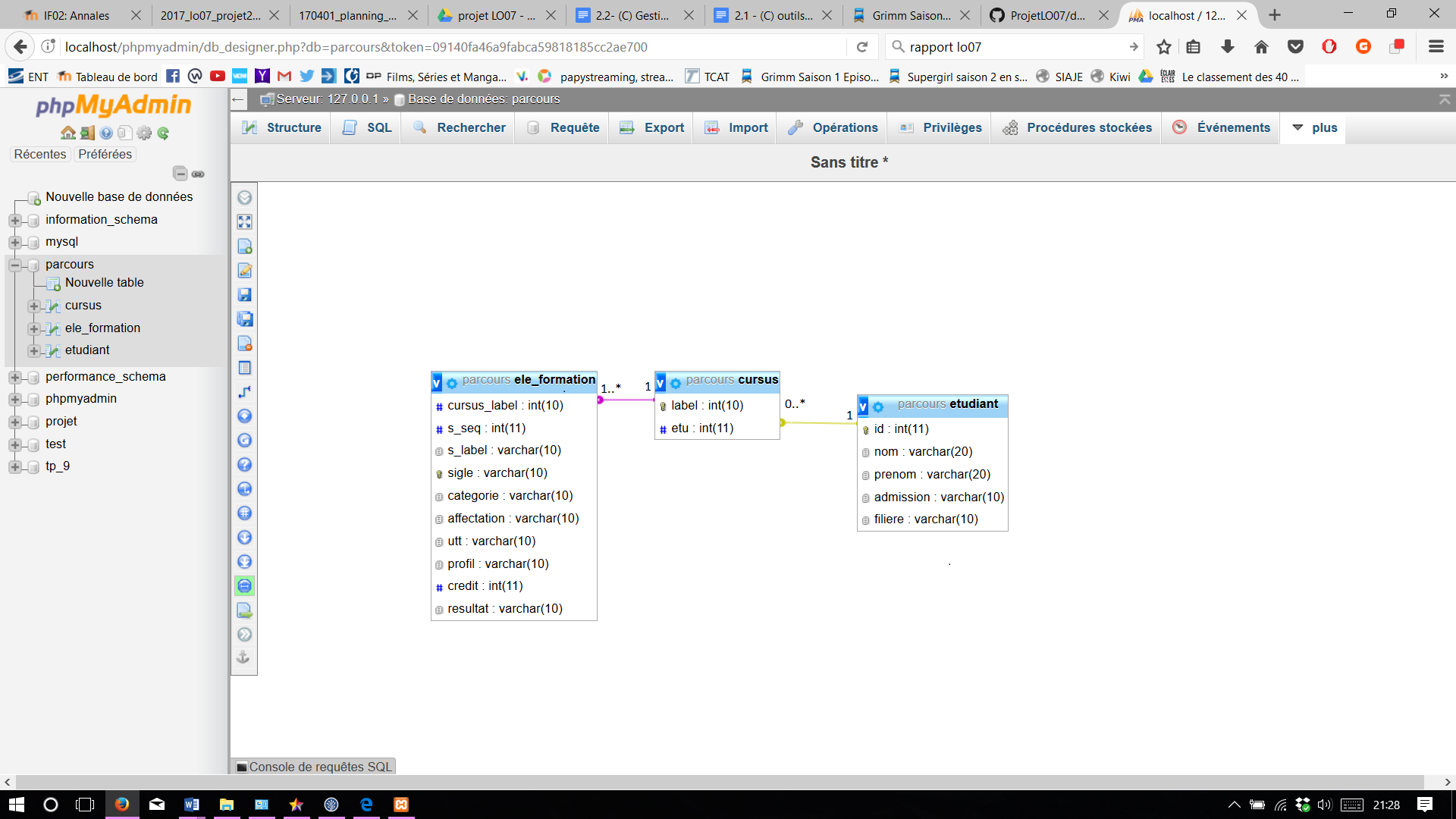
## Difficultés rencontrées

Pour des raisons de manque de temps ou de difficultés de développement trop énergivores à surmonter nous n’avons pas pu réaliser toutes les fonctionnalités du cahier des charges ou alors nous n’en avons réalisé certaines que partiellement.

Nous n’avons pas réussi à faire en sorte de pouvoir importer un nouveau règlement au format csv et vérifier si le cursus y est conforme. De plus la vérification de la conformité aux règlements ne se fait que pour le premier cursus de chaque étudiant, nous n’avons pas réussi à permettre à l’étudiant quel cursus il veut faire vérifier, lorsqu’il en a plusieurs. De même pour l’exportation, il n’est pas possible de choisir quel cursus l’étudiant veut importer, le site choisit le premier par défaut.  
Nous n’avons également pas eu le temps de permettre la duplication ou la modification des cursus déjà ajoutés.

# Etapes de conception de la base de données

## Schéma de la base de données



## Modèle physique des données

* **Table cursus :**

CREATE TABLE `cursus`

(

`label` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13,

`etu` int(1) NOT NULL,

PRIMARY KEY (`label`)

KEY `fk\_etu` (`etu`)

CONSTRAINT `fk\_etu` FOREIGN KEY (`etu`) REFERENCES `etudiant` (`id`),

)

* **Table ele\_formation :**

CREATE TABLE `ele\_formation`

(

`cursus\_label` int(10) DEFAULT NULL,

`s\_seq` int(11) NOT NULL,

`s\_label` varchar(10) NOT NULL,

`sigle` varchar(10) NOT NULL,

`categorie` varchar(10) NOT NULL,

`affectation` varchar(10) NOT NULL,

`utt` varchar(10) NOT NULL,

`profil` varchar(10) NOT NULL,

`credit` int(11) NOT NULL,

`resultat` varchar(10) NOT NULL,

PRIMARY KEY (`sigle`)

KEY `cursus\_label` (`cursus\_label`)

CONSTRAINT `fk` FOREIGN KEY (`cursus\_label`) REFERENCES `cursus` (`label`)

)

* **Table etudiant :**

CREATE TABLE `etudiant`

(

`id` int(11) NOT NULL,

`nom` varchar(20) NOT NULL,

`prenom` varchar(20) NOT NULL,

`admission` varchar(10) NOT NULL,

`filiere` varchar(10) NOT NULL,

PRIMARY KEY (`id`)

)

## Explications de la base de données

Une première table **« Etudiant »** va stocker toutes les données relatives aux étudiants de ISI. Ceux-ci ont donc un « id » qui est leur numéro d’étudiant, un « nom » et un « prénom », une « admission » qui précise quand l’étudiant a intégré l’UTT et une « filiere » qui précise la spécialisation en fin de branche. La clé primaire est ici l’ « id » puisque le numéro d’étudiant est forcément unique.

On a ensuite une table **« cursus »** qui possède un « label », c’est un numéro unique qui permet de l’identifier, c’est pourquoi c’est la clé primaire. Il est également défini par un « etu », qui correspond au numéro de l’étudiant qui suit ce cursus.

Enfin la table **« ele\_formation »** correspond à toutes les UE, les stages, les ateliers, documentaires, etc…, qui constituent un cursus. Chaque élément de formation a un « cursus\_label » qui correspond au label du cursus auquel il appartient. Il a également un attribut « s\_seq » qui correspond au numéro du semestre dans la formation de l’étudiant et un « s\_label » qui correspond au label du semestre (ex : ISI2). Un élément de formation est désigné par un « sigle » unique, qui sera la clé primaire. C’est ce sigle qui désigne l’élément et le nomme. L’attribut « categorie » désigne la catégorie d’UE ou d’élément comme par exemple CS, TM, ST, etc.. L’ « affectation » désigne le moment du parcours où l’élément a été suivi (TC, TCBR, FCBR), « utt » renseigne si l’ « élément a été suivi l’utt ou non, « profil » renseigne si l’élément appartient ou non au profil. Enfin chaque élément terminé est affecté d’un nombre de « crédits » obtenus et d’un « résultat » (A, B, C,…)

Un élément de formation appartient à un cursus qui est lui-même suivi par un étudiant. Les liaisons se font donc grâce aux contraintes de clé étrangères suivantes qui forment les jointures :

* L’attribut « etudiant » de la table **« Cursus »** référence l’ « id » de la table **« Etudiant »**.
* L’attribut « cursus\_label » de la table **« ele\_formation »** référence le « label » de la table **« cursus »**.

On a donc :   
cursus.etudiant = etudiant.id  
ele\_formation.cusus\_label = cursus.label

# Conclusion

Nous avons donc développé un site de gestion de cursus pour les étudiants. Celui-ci est relié à une base de données alimentée par les profils et les cursus des étudiants, qu’ils peuvent décrire par formulaire ou importer sous la forme d’un fichier csv. Ce cursus sont ensuite visualisables et l’étudiant peut vérifier s’il est conforme aux différents règlements des études de l’UTT. Un cursus enregistré dans la base de données peut également être exporté au format csv.

Les contraintes de développement (HTML, CSS, PHP, JavaScript, Base de données, serveur Apache) ont été respectées et ont permis le développement de la plupart des fonctionnalités.

Les points clés du projet ont été en particulier la gestion des formulaires dynamiques, devant s’adapter à n’importe quel nombre d’éléments de formation au sein d’un cursus, et la façon de relier les données collectées à la base de données. Ensuite la rédaction des fonctions d’évaluation des cursus par rapport aux règlements a également représenté une part prépondérante de ce travail.

Ce que ca nous a apporté :

CC

Comment on gere pages site avec base de données on gerer les donn2es dans la page, recup les donnees et stocker, faire lien bdd et site. Aws cloud server connaissances, config du cloud server, comment faire la connexion avec le serveur par ssh. Connaissances installations bdd sur serveur . Lire/ecrire les donnee dans le fichier CSV avec les fonctions de PHP

Delphine

…..

Efficacité github, repartir taches, partager mises a jour du code en temps reel, pb