

Rapport Final

Table des matières :

I - Analyse et conception

- a) Cahier des charges
- b) Organisation
- c) Modèle conceptuel
- d) Présentation des tables
- e) Modèle relationnel
- f) Requêtes de création de table

II - Documentation technique

- a) Guide d'utilisation
- b) Schéma de l'organisation des fichiers
- c) Explication des requêtes difficiles

I - Analyse et conception

a) Cahier des charges

Un établissement est venu nous voir car, chaque année, chaque directeur de spécialité crée un fichier personnel dans lequel il gère les affectations enseignant-enseignement (il sait quel enseignant enseigne quelle matière). Or le problème est qu'il n'y a pas de plateforme commune dans laquelle chaque directeur de spécialité pourrait voir le service de ses enseignants intervenant dans plusieurs spécialités. En effet, il est difficile pour le directeur d'organiser les affectations sans savoir quand ses enseignants donnent cours dans les autres spécialités. De plus, le directeur de formation, en charge d'une spécialité, doit organiser la répartition des heures de cours en respectant les quotas imposés :

- Un enseignant doit faire au moins 384 heures par an
- Un enseignant chercheur doit faire minimum 192 heures par an
- Un a.t.e.r doit faire exactement 192 heures par an
- Un doctorant assistant doit faire minimum 64h par an
- Un vacataire doit faire au moins 96h par an

De plus, ces quotas sont donnés en "équivalent TD". En effet, une heure de cours magistral (CM) vaut 1,5 h de travaux pratiques (TD / TP) et un CTD vaut 1,25 h de travaux pratiques (TD / TP).

Comme dit précédemment, à l'heure actuelle, chaque directeur planifie manuellement les maquettes et le manque de centralisation rend la prise de décisions budgétaires difficile.

Cette application permettra de résoudre le problème de manque de centralisation en organisant les 10 spécialités disponibles à Polytech. Elle permettra au directeur de chaque spécialité une gestion plus adéquate des affectations enseignant-enseignement en pouvant ajouter et affecter **de manière simple** un enseignant à une spécialité . De plus, le client nous a demandé une visualisation des maquettes et des affectations. L'application permettra donc au directeur de formation d'avoir une vision globale de toutes les unités d'enseignement présentes dans sa spécialité ainsi que les modules (avec les quotas horaires de ces modules) présents dans chacune de ces unités d'enseignement, tout ça au même endroit.

L'application permettra à chaque directeur de formation d'avoir accès aux quotas horaires des enseignants intervenant dans sa spécialité et permettra également à chaque enseignant

de voir son propre service. Chaque enseignant pourra facilement filtrer son service prévisionnel par spécialité / type (cours magistral, travaux dirigés ou travaux pratiques) / semestre. Il pourra, en plus, avoir accès à son propre quota horaire. Cette application a aussi pour but d'avoir une vision globale des coûts liés aux enseignants, à savoir combien d'heures supplémentaires a besoin chaque spécialité afin d'aider la direction à prendre des décisions budgétaires.

Pour affecter un enseignant à un module, il faudra que l'application demande d'abord à l'utilisateur le nom de la spécialité, l'année et le semestre et, en fonction de ce choix, il devra sélectionner le nom du ou des professeur(s) réalisant chaque module de cette spécialité (le tout sous forme de listes). En effet, le nombre d'enseignants à sélectionner dépend du type de cours donné : si c'est un CM il n'y a qu'un groupe (par défaut), si c'est un TD il y a maximum deux groupes donc deux professeurs, et si c'est un TP il y a maximum trois groupes donc maximum trois professeurs.

En outre, cette application permettra de :

Gérer les enseignants

- ajouter / modifier un enseignant
- stocker son statut (enseignant, enseignant chercheur, a.t.e.r, doctorant assistant ou vacataire)
- Consulter les enseignants présents dans l'établissement en ayant une visualisation détaillée des heures effectuées par chaque enseignant.

Gérer les formations

- 10 spécialités
- leurs unités d'enseignement (UE)
- leurs modules
- leurs types (CM, TD, TP, CTD)
- leurs volumes horaires prévus

Affecter les enseignants à un/des module(s)

- lier *un enseignant* à *un module*
- indiquer le type de cours que le professeur donne (CM/TD/TP/CTD)

Visualiser

- la maquette d'une spécialité

- les affectations des différents modules
- le service d'un enseignant (avec des filtres)
- le service global d'une spécialité

Périmètre

- On ne se penchera pas sur l'administration des droits, on considère que l'utilisateur effectue ce qu'il est seulement autorisé à faire.
- Nous ne nous pencherons pas sur les études budgétaires en chiffres.

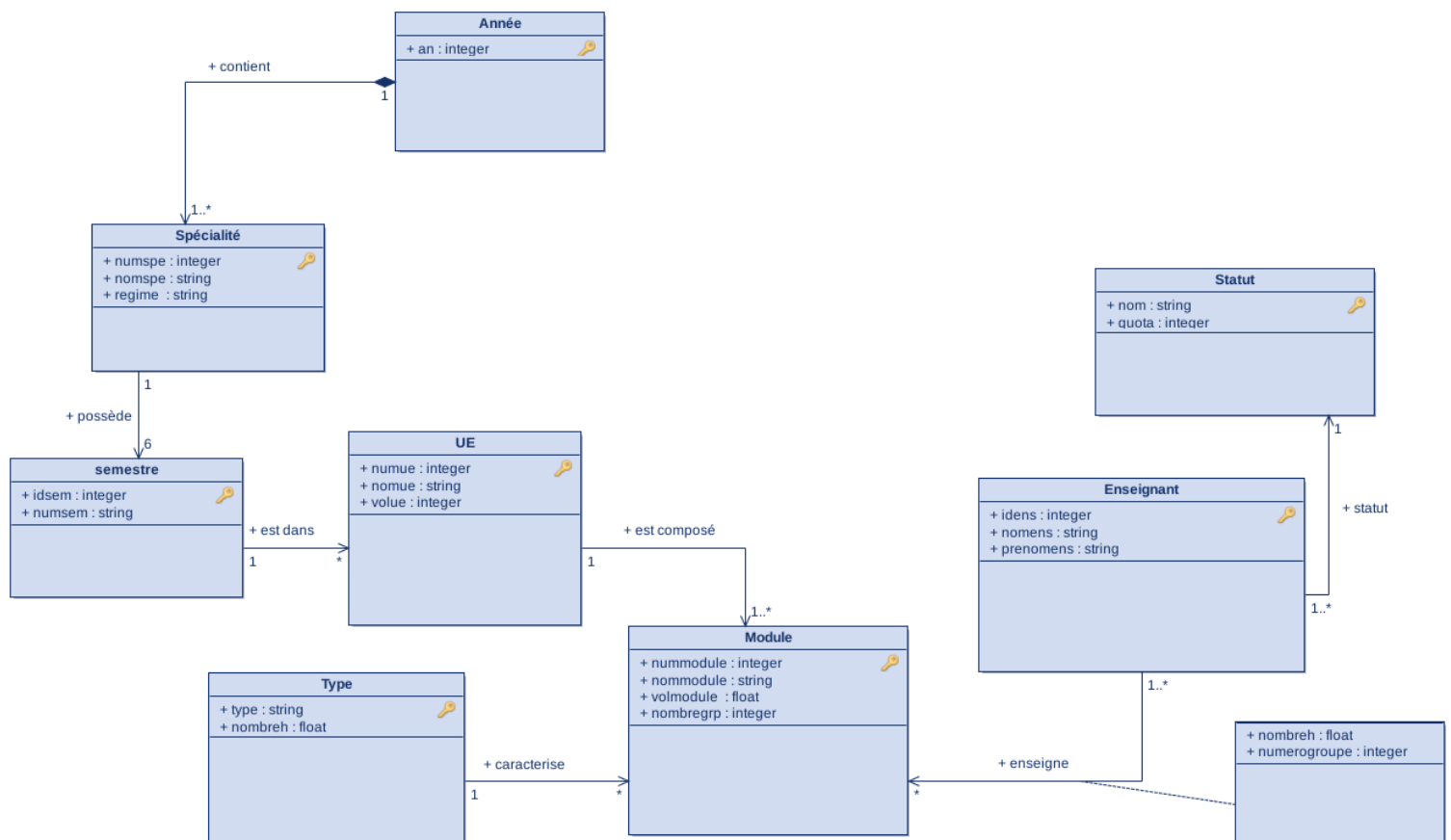
b) Organisation :

Le projet se déroule sur 22h réparties ainsi :

- 4h pour l'analyse du projet, la répartition des tâches et l'élaboration du cahier des charges.
- 18h sur la conception de l'application.

Pour la répartition du travail, mon binôme s'est occupé de la création de la base de données ainsi que de son remplissage et je me suis occupé de coder les premières pages du site. À la suite de cela, nous nous sommes répartis les différentes pages du site web: j'ai par exemple codé la page de consultation des enseignants pendant que mon binôme a codé la page de visualisation de la maquette.

c) Modèle conceptuel :



Justification:

L'application demandée par le client doit permettre de gérer et d'élaborer les services prévisionnels des enseignants. Avant tout, notre schéma conceptuel doit permettre de représenter l'ensemble des éléments nécessaires au fonctionnement d'un établissement comme Polytech, afin que l'application puisse gérer de manière cohérente les spécialités, les unités d'enseignement, les modules et les services des enseignants. Pour cela, nous avons choisi d'inclure les classes :

- **Spécialité** : elle nous permet d'enregistrer toutes les spécialités de Polytech
- **UE** : elle nous permet de savoir facilement de quelles unités d'enseignement est composé une spécialité donnée
- **Module** : elle nous permet de savoir de quels enseignements est composé chaque UE
- **Enseignant** : cette classe nous permet d'enregistrer tous les enseignants présents à Polytech et nous permet également (comme demandé par le client) d'ajouter de nouveaux enseignants
- **Statut** : Cette classe nous permet de connaître le statut d'un enseignant (enseignant, enseignant chercheur, a.t.e.r, doctorant assistant ou vacataire), elle nous permet aussi de connaître le quota horaire par statut; comme dit précédemment, chaque statut ne doit pas remplir le même quota horaire.

En plus de ces quatre classes principales, il est nécessaire d'en rajouter afin de répondre aux exigences du client :

L'application doit permettre aux enseignants de connaître précisément les détails de son service, autrement dit un enseignant doit pouvoir avoir accès sans problème à son quota horaire total selon différents filtres (il peut voir le service total dans chaque formation/spécialité/ type/ semestre). Pour faire cela, nous avons créé une classe association entre la table Enseignant et la table Module. Cette classe association a pour attribut le type qui permettra à l'enseignant de filtrer par type (CM,TD,TP,CTD) et un nombre d'heures qui comptera les heures données par cet enseignant. Pour afficher le quota horaire total de chaque enseignant, nous utiliserons une requête php simple qui sommerá toutes les heures effectuées par l'enseignant. Pour ce qui est du filtrage selon la formation, il nous suffira de sélectionner la spécialité voulue dans la table spécialité. Pour finir, pour le filtrage selon le semestre, nous avons fait le choix d'ajouter une classe **Semestre** nous permettant de choisir le numéro du semestre voulu.

Pour visualiser le service total d'un enseignant, il nous suffira de ne filtrer que par le numéro de l'enseignant voulu et d'afficher la somme de toutes les heures qu'il a faites pour tous les types de cours mélangés. Typiquement : heures totales= heures de CM + heures de TD + heures de TP (⚠ il faudra prendre en compte le fait qu'une heure de CM ou de TP se donne en équivalent TD avec 1h de CM=1,5h de TD et 1h de CTD=1,25h de TD).

Le client nous a également demandé de pouvoir avoir un historique par année de ces données, c'est pour cela que nous avons créé la classe **Annee** qui nous permettra de filtrer les données par année voulue.

Enfin, pour visualiser la maquette d'une spécialité, il nous suffira d'afficher toutes les UE de cette spécialité et tous les modules correspondant à chacune de ces UE.

Les cardinalités :

- **Annee <-> Specialite (composition) : 1-1..***
Une année possède au moins une spécialité et une spécialité correspond à une année.
"Composition" car une spécialité n'existe pas sans son année.
- **Specialite <-> Semestre : 1-6**
Une spécialité a six semestres mais un semestre ne correspond qu'à une spécialité.
- **Semestre <-> UE: 1-***

Une UE ne fait référence qu'à un semestre mais un semestre contient plusieurs UE.

- **UE <-> Module : 1-1..***

Un module ne peut être que dans une seule UE mais une UE est composée d'au moins un module (sinon l'UE n'aurait pas lieu d'exister).

- **Type <-> Module : 1-***

Cette table nous permet d'afficher la maquette d'une spécialité et donne le nombre d'heures prévues en CM, TD, TP, CTD pour un module donné. Ainsi, un type fait référence à plusieurs modules mais un module est caractérisé par un type.

- **Enseignant <-> Module : 1..*-***

Un enseignant enseigne dans plusieurs modules et un module a un ou plusieurs enseignant(s) (si aucun enseignant n'est affilié au module alors celui-ci ne peut être assuré et donc ne peut exister).

- **Statut <-> Enseignant : 1-1..***

Un enseignant possède un statut et un statut correspond à au moins un enseignant (sinon ce statut n'a pas lieu d'exister).

d) Présentation des tables

Table Année	Nom Attribut	Type	Justification	Exemple
Clé Primaire	an	Int (>0 et <10000)	Année correspondant à la première du cycle	2024 (pour 24/25)

Table Spécialité	Nom Attribut	Type	Justification	Exemple
Clé Primaire	numspe, an	Int Int (année)	Relation maître esclave : une spé n'existe pas sans son année	14,2024

Attributs	nomspe	string[100]	Un nom caractérisant chaque spécialité	“Ingénieur Informatique, Statistique et Intelligence Artificielle”
	regime	string[10]	Alternance ou Continu	“Continu”

Table Semestre	Nom Attribut	Type	Justification	Exemple
Clé Primaire	idsem	Int	identifiant du semestre	5
Attributs	numsem	string	numéro du semestre	S5
	numspe,an	Int Int (année)	Clé étrangère de la table spécialité, 1 idsem = 1 spé,an	14,2024

Table UE	Nom Attribut	Type	Justification	Exemple
Clé Primaire	numue	Int	Numéro unique pour chaque UE	7
Attributs	nomue	string[100]	Le nom de l'UE pour l'utilisateur	“Algorithme et Programmation”
	volue	Int	Nombre d'heures correspondant à la somme des volmodules	160

Table Module	Nom Attribut	Type	Justification	Exemple
Clé Primaire	nummodule	Int	Un numéro unique pour différencier chaque module pouvant avoir le même nom (ex : info en MECA	15

			<i>et en GBA)</i>	
Attributs	nommodule	string[100]	Le nom du module pour l'utilisateur	"Algorithme et Programmation"
	volmodule	Int	Nombre d'heures du module	160
	nombrgrp	Int (>0 et <4)	Nombre de groupes qui dépendent du type de cours	3

Table Type	Nom Attribut	Type	Justification	Exemple
Clé Primaire	type	string[2]	Si le cours est un CM, un TD, un TP ou un CTD	"TD"
Attribut	nombreh	Int (en heures) (>0 et <1000)	Le nombre d'heures prévisionnel	50h

Table Enseignant	Nom Attribut	Type	Justification	Exemple
Clé Primaire	idens	Int (>0 et <100000)	Un numéro unique pour différencier chaque enseignant	1
Attributs	nomens	string[100]	Le nom de famille de l'enseignant	"xxxx"
	prenomens	string[100]	Le prénom de l'enseignant	"xxxxx"

Table Statut	Nom Attribut	Type	Justification	Exemple
Clé Primaire	nom	string[100]	Définit le statut de l'enseignant	"Enseignant chercheur"
Attribut	Quota	Int (en heures) (>0 et <1000)	Le nombre d'heures max/min qui change selon le statut	192 h

Table Enseigne	Nom Attribut	Type	Justification	Exemple
Clé Primaire	idens, nummodule	Int (>0 et <100000)	Définit le statut de l'enseignant	9 16
Attributs	nombreh	Int (en heures) (>0 et <1000)	Le nombre d'heures que l'enseignant dispense	24
	numerogrp	int (>0 et <4)	le numéro du groupe de TD et TP	2

Nous avons optimisé notre schéma en ne mettant que les relations nécessaires. C'est pourquoi il faudra faire une jointure entre module et UE puis specialite pour avoir accès à tous les modules de celle-ci. Nous avons réalisé ce schéma conceptuel à l'aide du logiciel modelio. L'attribut statut précise si l'enseignant est chercheur, vacataire ou doctorant. L'attribut volue n'est que la somme de tous les volmodule, soit le volume horaire total d'une UE. L'attribut regime précise si la formation est continue ou en alternance.

e) Modèle relationnel :

Specialite (numspe, #an, nomspe, regime)

Enseignant(idens, nomens, prenomens, #nom)

Statut(nom, quota)

UE (numue, nomue, volue, #idsem)

Module (nummodule, #type, nommodule, volmodule, nbrgrp, #numue)

Type (type)

Enseigne (#idens,#nummodule, nbheure, numerogroupe)

Semestre (idsem, numsem, #numspe, #an)

Annee(an)

Justification :

La clé de l'UE est composée du numéro du semestre et d'un numéro qui lui est propre.

La classe Specialite est régie par une composition, elle n'existe pas sans son année. Dans la classe Enseigne, l'attribut nbheure correspond au volume des modules qui seront affectés aux enseignants.

f) Requêtes de création de table

À l'aide du modèle relationnel, nous pouvons écrire les requêtes de création de table sous Postgres. Ces commandes sont écrites avec la syntaxe spécifique à Postgres. Ces requêtes découlent du modèle relationnel.

```
CREATE TABLE annee(  
  an INTEGER,  
  PRIMARY KEY(an));
```

“an” est la clé primaire de la table Annee. En effet, une année est unique et donc cela peut bien être une clé primaire.

```
CREATE TABLE specialite(  
  numspe INTEGER,  
  nomspe VARCHAR(100),  
  regime VARCHAR(10),  
  an INTEGER,  
  FOREIGN KEY(an) REFERENCES annee(an),  
  PRIMARY KEY(numspe, an));
```

Chaque spécialité est rattachée à un numéro unique “numspe”, c’est donc la clé primaire de cette table. Il faut ensuite le nom de chaque spécialité, c’est pour cela que nous avons un attribut “nomspe” qui n'excédera jamais 100 caractères (nous avons fait exprès de prévoir large au cas où il y aurait une spécialité avec un nom assez long). Il faut ensuite stocker le régime de la spécialité (continu, alternant) et, comme dit dans la partie “cardinalité”, il y a une composition entre Annee et Specialite, d’où la présence de “an” à la fois en clé primaire et en clé étrangère.

```
CREATE TABLE semestre(  
  idsem INTEGER,  
  numsem VARCHAR(3),  
  numspe INTEGER,  
  an INTEGER,  
  FOREIGN KEY(numspe, an) REFERENCES specialite(numspe, an),  
  PRIMARY KEY(idsem));
```

Chaque semestre doit avoir son propre identifiant, c’est pour cela que l’on a “idsem” en clé primaire et “numsem” en attribut. En effet, si on avait décidé de mettre le numéro de semestre en clé primaire, nous n’aurions pas pu affilier deux spécialités différentes au même

semestre. Nous avons donc fait le choix d'identifier chaque semestre par un identifiant. Chaque semestre est relié à une spécialité, c'est pour cela que l'on a "numspe" en clé étrangère.

```
CREATE TABLE ue(  
  numue INTEGER,  
  nomue VARCHAR(100),  
  volue FLOAT,  
  idsem INTEGER,  
  FOREIGN KEY(idsem) REFERENCES semestre(idsem),  
  PRIMARY KEY(numue));
```

Chaque UE a un numéro lui servant d'identifiant unique. Il faut aussi stocker le volume de chaque UE (il est ici en *float* car ce volume est rentré en équivalent TD, il conviendra ensuite de l'afficher en heures effectives via une requête php). Chaque UE est reliée à des semestres, c'est pour cela que "idsem" est placé en tant que clé étrangère.

```
CREATE TABLE type(  
  type VARCHAR(2),  
  nombre INTEGER,  
  PRIMARY KEY(type));
```

Il faut ensuite créer la table type qui renseigne sur le type d'un module (CM,TP,TD,CTD). Nous avons ici un type avec deux caractères. En effet, dans notre modèle, CTD est stocké sous 'CT' (il faudra donc une requête php pour afficher CTD quand CT est sélectionné).

```
CREATE TABLE module(  
  nummodule INTEGER,  
  nommodule VARCHAR(100),  
  volmodule INTEGER,  
  nbrgrp INTEGER,  
  numue INTEGER,  
  type VARCHAR(2),  
  FOREIGN KEY(numue) REFERENCES ue(numue),  
  FOREIGN KEY(type) REFERENCES type(type),  
  PRIMARY KEY(nummodule, type));
```

Chaque module est identifié par un "nummodule" unique placé en tant que clé primaire. On a aussi placé "nommodule" en attribut de 100 caractères. Comme chaque module est relié à une UE, "numue" est en clé étrangère. De même, chaque module possède un type (CM,CTD,TP,TD), c'est donc pour cela que l'identifiant de la classe type est placé en clé étrangère.

```
CREATE TABLE statut(  
  nom VARCHAR(100),  
  quota INTEGER,  
  PRIMARY KEY(nom));
```

Le nom du statut est stocké en clé primaire. Le quota horaire imposé pour chaque statut est placé en tant qu'attribut.

```
CREATE TABLE enseignant(  
  idens INTEGER,  
  nomens VARCHAR(100),  
  prenomens VARCHAR(100),  
  nom VARCHAR(100),  
  PRIMARY KEY(idens),  
  FOREIGN KEY(nom) REFERENCES statut(nom));
```

Chaque enseignant possède un identifiant "idens" personnel unique (c'est donc la clé primaire), un nom et un prénom qui n'excèdent pas 100 caractères (nous avons pensé aux personnes avec de longs nom/prénom). Le nom du statut est ici placé en tant que clé étrangère.

```
CREATE TABLE enseigne(  
  idens INTEGER,  
  nummodule INTEGER,  
  numerogroupe INTEGER,  
  nbheure INTEGER,  
  FOREIGN KEY(idens) REFERENCES enseignant(idens),  
  FOREIGN KEY(nummodule) REFERENCES module(nummodule),  
  PRIMARY KEY(idens, nummodule));
```

Cette table est une classe association entre enseignant et module, c'est donc pour cela qu'elle possède "idens" et "nummodule" à la fois en clé primaire et en clé étrangère. Elle possède "nbheure" comme attribut pour savoir combien d'heures l'enseignant a effectué dans chaque cours.

II - Documentation technique

a) Guide d'utilisation



Voici la première page de notre site, la page d'accueil, on y trouve 3 sections :

- Édition : permet aux directeurs de spécialités d'affecter les différents enseignants à un module, ajouter des enseignants et d'en supprimer.
- Consulter une maquette : permet à l'utilisateur de consulter la maquette d'une spécialité.
- Consulter la liste des enseignants : permet à l'utilisateur de visualiser la liste de tous les enseignants présents dans l'établissement avec quelques informations supplémentaires et, s'il le souhaite, voir le détail du service d'un enseignant.

Bienvenue dans le mode d'édition

Affectations

Ajouter un
enseignant

Supprimer
un
enseignant

Cliquons sur la première section : Édition.

Lorsque l'utilisateur clique sur "Édition", il arrive sur cette page, celle-ci permet :

- d'affecter un enseignant à un module
- d'ajouter/supprimer un enseignant de l'établissement

Cliquons sur : Affectations

Veuillez sélectionner vos filtres :

Spécialité : IS2A Année : 2025 Semestre : SS OK

Accueil

L'utilisateur est alors invité à remplir les trois filtres de sélection : la spécialité voulue, l'année et le semestre. Il clique ensuite sur "OK" (un bouton accueil permet le retour à la page d'accueil si besoin)

Veillez sélectionner vos filtres :

Spécialité : Année : Semestre :

Résultats pour : GBA - 2025 - S5

Module	Volume (Type)	G1	G2	G3
Automatique	22 h (CM)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Automatique	13 h (TP)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Automatique	14 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Biochimie	4 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Biochimie	30 h (TP)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Biochimie	30 h (CM)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
English for engineers	24 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Introduction au génie des Procédés	16 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Introduction au génie des Procédés	20 h (CM)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Microbiologie générale	24 h (CM)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Microbiologie générale	31 h (TP)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Microbiologie générale	2 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Statistiques	24 h (TD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Statistiques	18 h (TP)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>
Statistiques	13 h (CTD)	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>	<input type="text" value="Enseignant"/>

Il arrive alors sur cette page sur laquelle il peut :

- consulter les affectations par module
- affecter un ou plusieurs enseignant(s) (en fonction du type de cours : CM, TP, TD, CTD)

Il clique sur "valider" pour valider les affectations.

Si l'utilisateur souhaite maintenant ajouter un enseignant dans l'établissement, il clique sur "Ajouter un enseignant" sur la page d'accueil du site.

Ajouter un enseignant

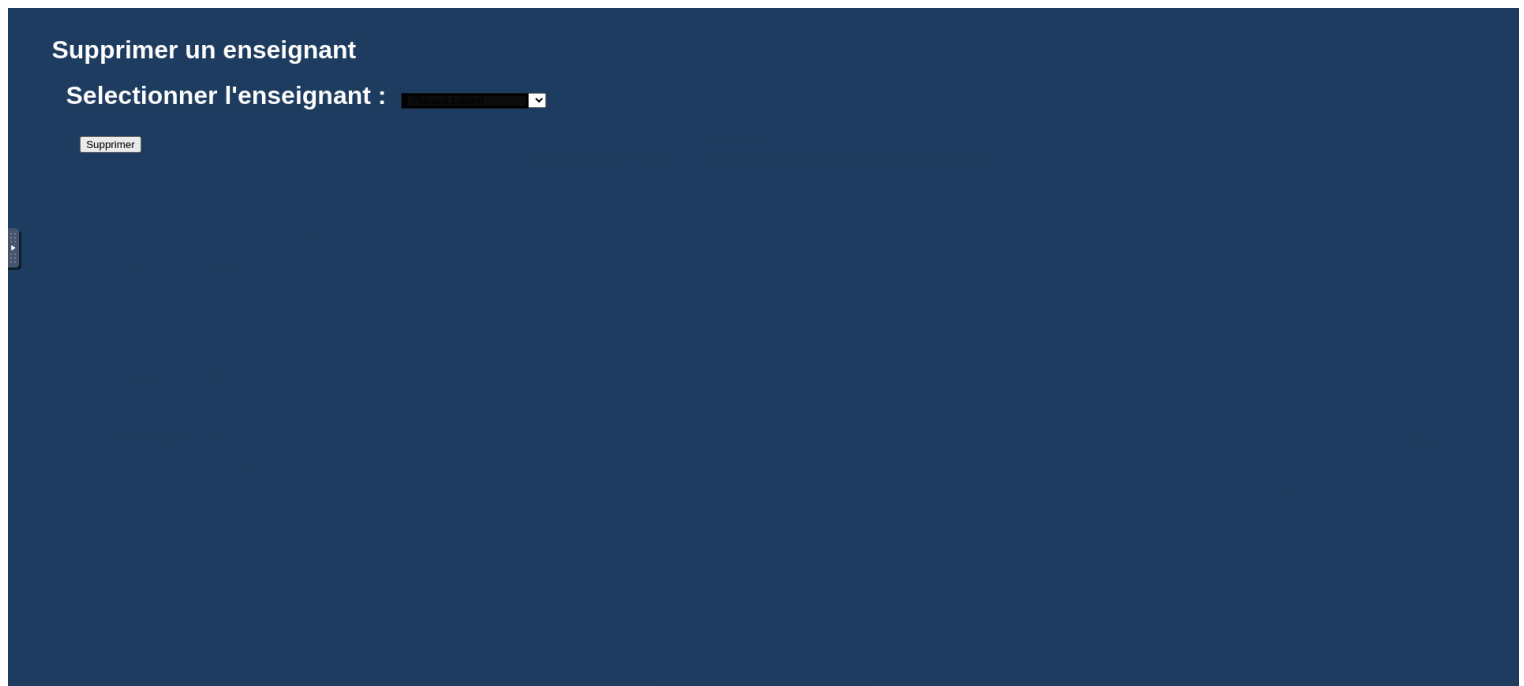
Prénom de l'enseignant :

Nom de l'enseignant :

Statut de l'enseignant :

Il arrive alors sur cette page où il est invité à remplir le prénom, le nom et le statut de cet enseignant. Une fois fait, il clique sur “ajouter” et l’enseignant est automatiquement ajouté à l’établissement.

De même, pour supprimer un enseignant, l'utilisateur clique sur “Supprimer un enseignant” sur la page d’édition.



Quand l'utilisateur arrive sur la page de suppression d'enseignant, il n'a qu'à sélectionner dans le menu déroulant l'enseignant qu'il souhaite supprimer. Cette action entraînera la suppression de l'enseignant ainsi que toutes les données relatives à cet enseignant sur le site.

Si l'utilisateur souhaite consulter une maquette, il sélectionne “consulter une maquette” sur la page d'accueil

Spécialité :
Spécialité ▼

Régime :
Régime ▼

Année :
Année ▼

OK

L'utilisateur est alors invité à sélectionner la spécialité, le régime (alternant ou continu) et l'année de la maquette qu'il souhaite consulter.

Dans l'exemple suivant, on considère que l'utilisateur a choisi GBA-continu-2025

Résultats pour : GBA - continu - 2025 (Total : 895.5 h)

Unités d'enseignement (UE)

Nom	Volume horaire
Introduction au génie biologique	121
Outils de l'ingénieur GBA 1	126.5
Langues	24
Génie biologique	58
Développement et alimentation durables 2	44
Langues	24
Génie des procédés	54.5
Génie biologique et alimentaire	107
Langues	24
Génie industriel	75
Biotechnologies industrielles	77
Langues	24
Management Qualité et Environnement	60.5
Economie circulaire	76

Accueil



L'utilisateur obtient alors un tableau résumant l'ensemble des Unités d'enseignement (avec le volume horaire par Unité d'enseignement et le volume horaire total) présentes dans la spécialité sélectionnée. Si l'utilisateur veut en savoir plus sur une unité d'enseignement et souhaite le détail des modules composant une Unité d'enseignement, il peut cliquer sur le nom de l'Unité d'enseignement dont il veut des détails et arrive sur cette page :

Modules de l'UE 1

Numéro module	Nom du module	type	nombre de groupes	Volume horaire
1	Biochimie	CM	1	30
2	Biochimie	TD	2	4
3	Biochimie	TP	3	30
4	Microbiologie générale	CM	1	24
5	Microbiologie générale	TD	2	2
6	Microbiologie générale	TP	3	31

L'utilisateur voit ici l'ensemble des modules présents dans l'Unité d'enseignement sélectionnée. Il y voit le nom des modules détaillés par type et peut y consulter le volume horaire de chaque module/type. Par exemple, ici, il y a 30h dédiées aux cours magistraux de Biochimie.

Enfin, si l'utilisateur souhaite consulter l'ensemble des enseignants présents dans l'établissement, il doit cliquer sur "consulter la liste des enseignants" sur la page d'accueil du site.

Enseignants présents dans l'établissement

Nom	Statut	Quota	Service actuel	Etat
Alexandre Burlet	vacataire	96	0	Sous-Service (-96)
Cristian Preda	vacataire	96	0	Sous-Service (-96)
Daniilo Rodriguez de Lima	doctorant assistant	64	0	Sous-Service (-64)
Franck Seynhaeve	enseignant	384	4.5	Sous-Service (-379.5)
François Boulier	enseignant	384	0	Sous-Service (-384)
Marc Vasseur	enseignant	384	0	Sous-Service (-384)
Marie-Eléonore Kessaci	enseignant	384	0	Sous-Service (-384)
Mélanie Momont	enseignant	384	0	Sous-Service (-384)
Nathalie Devesa	enseignant	384	0	Sous-Service (-384)
Rosane Ushirobira	vacataire	96	100	Heures Supp (4)
Sofiane Maazi	vacataire	96	96	ok ()
Stéphane Janot	enseignant	384	0	Sous-Service (-384)
Thibault Liétard	enseignant	384	3	Sous-Service (-381)

L'utilisateur voit donc l'ensemble des enseignants présents dans l'établissement. Des informations supplémentaires sont données par enseignant comme son statut, son quota annuel, son quota effectif (ces derniers sont donnés en équivalent TD) et son état : si l'enseignant a fait moins d'heures que son quota il sera sous l'état "sous service", s'il a fait exactement le nombre d'heures qu'il devait faire il sera sous l'état "ok" (comme Sofiane Maazi dans l'exemple), enfin s'il a fait plus d'heures que nécessaire il sera sous l'état "Heures supp" avec le nombre d'heures qu'il a faites en plus entre parenthèses.

Si l'utilisateur souhaite avoir le détail de son service, il lui suffit simplement de cliquer sur son nom.

Votre quota horaire (donné en équivalent TD) : 3/384h

Votre service en détail :

Spécialité : semestre : Type :

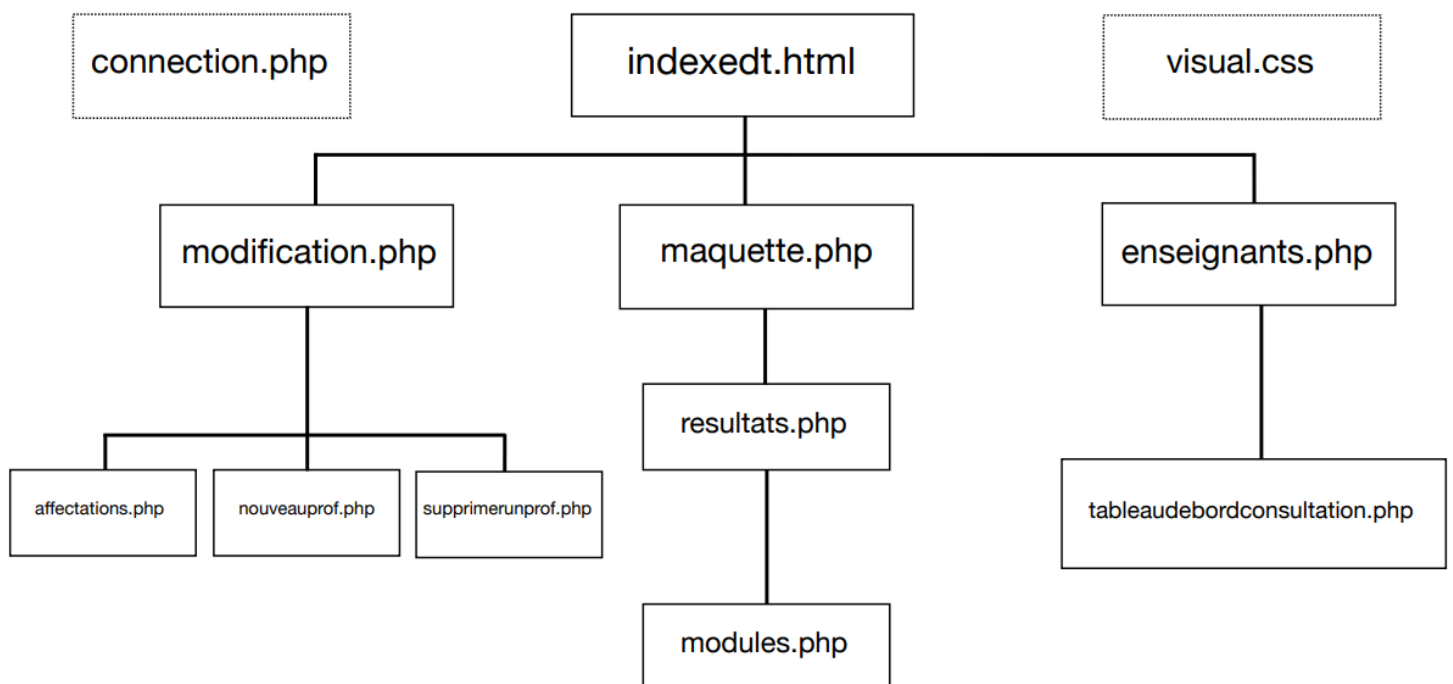
Spécialité	Module	Type	Nombre d'heures
GBA	Biochimie	TD	3

**Liétard
Thibault
enseignant**

Si l'utilisateur clique sur son nom, il arrive alors sur sa page personnelle. Il peut y consulter différentes informations le concernant :

- Le rapport entre son quota horaire actuel et son quota annuel fixé par son statut (dans l'exemple ci-dessus, M.Liétard est un enseignant et doit donc faire au moins 384h par an).
- Il peut consulter son service en détail : dès son entrée sur cette section du site, ce dernier affiche toutes les spécialités dans lesquelles il intervient cette année. Il y voit en plus le détail des différents modules, les types de cours qu'il donne et son nombre d'heures par spécialité/module/type. Il peut bien sûr filtrer les résultats par spécialité, semestre et type pour n'afficher que son service suivant ce qu'il souhaite consulter.

b) Schéma de l'organisation des fichiers



Pages reliées à toutes les autres sauf `connection.php` qui est relié à toutes les autres sauf `indexedt.html`

c) Explication des requêtes difficiles :

Dans la page nouveauprof.php, il nous fallait trouver un moyen de créer un identifiant unique à chaque ajout d'un enseignant. Il fallait que cela soit fait en interne pour ne pas contraindre l'utilisateur qui ajoute le nouvel enseignant à trouver un identifiant non existant. Pour cela, on a décidé de récupérer tous les identifiants existants, de récupérer le maximum des identifiants et d'ajouter 1 à chaque ajout de professeur. La requête pour faire cela est :

```
$sqlidentifiant="select max(idens) as id from enseignant";  
$id=$ligneidentifiant['id']+1;
```

Dans la page enseignants.php, il faut sélectionner l'identité du professeur (prénom+nom), son statut, le quota horaire qu'il doit réaliser en fonction de son statut (c'est pour cela qu'on demande de sélectionner statut.quota). On voulait ensuite afficher l'ensemble des ces heures effectives, c'est donc pour cela que l'on somme ses heures dans la classe enseigne (sum(enseigne.nbheure)). On effectue ensuite un "left join" car on veut aussi récupérer les enseignants qui n'ont pas encore fait d'heures car le but de cette page est d'afficher l'ensemble des enseignants présents dans l'établissement donc il faut aussi afficher ceux qui n'ont pas encore d'heures d'enseignement à leur actif. On a enfin choisi de les afficher dans un ordre croissant d'où le "order by nom asc". Voici la requête en question :

```
$sql="select enseignant.idens, enseignant.prenoms||' '||enseignant.nomens as nom,  
statut.nom as nomstatut,statut.quota, sum(enseigne.nbheure) as serviceactuel from  
enseignant left join statut on statut.nom=enseignant.nom left join enseigne on  
enseigne.idens=enseignant.idens group by  
enseignant.idens,enseignant.nomens,enseignant.prenoms,statut.quota,statut.nom order  
by nom asc";
```

Dans la page affectations.php, Il fallait un moyen pour afficher l'enseignant dans le cas où il est déjà affecté sur le module. S'il l'est, il faut l'afficher en valeur par défaut et ne pas l'afficher dans le tableau déroulant. Si l'on n'y touche pas, le formulaire traitera cette sélection comme les cas où nous avons laissé la valeur par défaut "enseignant".

```
// Vérifie si pour un module et un groupe donné la valeur existe  
$idens_affecte = $affectations[$mod['nummodule']][$g] ?? null;  
// Si oui, on récupère son nom et prénom et on affiche en option par défaut :  
if ($idens_affecte)  
{  
    $res_ens_nom = pg_query("SELECT prenomens || ' ' || nomens AS nom  
FROM enseignant WHERE idens = $idens_affecte");  
    $ens_nom = pg_fetch_result($res_ens_nom, 0, 0);  
    echo "<option value='$idens_affecte' selected>$ens_nom</option>";  
}
```

```

// Affichage des autres enseignants disponibles
pg_result_seek($res_ens, 0);
while($ens = pg_fetch_assoc($res_ens))
{
// Ne pas afficher l'enseignant déjà sélectionné
if ($ens['idens'] != $idens_affecte)
{
echo "<option value='{ $ens['idens'] }'>{ $ens['nom'] }</option>";
}
}

```

Dans la page affectations.php, dans le cas où l'on veut modifier une affectation qui existe, il faut créer une valeur booléenne 'exists' qui, si vraie, supprime celle qui existe déjà. Il y a donc une requête pour supprimer l'affectation dans la table enseigne. Une fois cela fait ou si elle n'existe pas, on insère l'affectation.

```

//Création du booléen
$exists = pg_num_rows($res_check) > 0;
//Si oui
if ($exists)
{
//On supprime l'affectation existante
pg_query($con, "
DELETE from enseigne
WHERE nummodule = $nummodule and numerogroupe = $numgroupe
");
}
//On affecte la valeur dans la table enseigne en ayant récupéré le volume du module
$res = pg_query($con, "SELECT volmodule FROM module WHERE nummodule =
$nummodule");
$vol = pg_fetch_result($res, 0, 0);
// Mise à jour de l'affectation
pg_query($con, "
insert into enseigne values($idens, $nummodule,$numgroupe, $vol)
");
}

```