

RÉALISATION D'UN SYSTÈME DE GESTION D'ABSENCES DE POLYTECH



Résumé :

Polytech Annecy-Chambéry possède des compétences nombreuses et diverses en informatique. Toutefois, elle ne possédait pas de système permettant de suivre activement la présence des étudiants aux différents cours. Actuellement, cette vérification est partielle et occasionnelle. Ainsi l'absence des étudiants n'est pas systématiquement relevée pour plusieurs raisons pratiques.

Les méthodes manuelles actuelles sont longues et pénibles autant de la part des enseignants que de la part de la scolarité. Ce travail peut prendre des heures si chaque étudiant était suivi.

Le projet actuel propose de faciliter l'ensemble du système de gestion des absences de Polytech par l'élaboration d'une base de données qui relie les informations entre les étudiants, les enseignants, les cours et les modules, tout en enregistrant la présence de chaque étudiant à chacun de ses cours respectifs.

Cette base de données sera utilisée afin d'imprimer les feuilles d'absences, qui seront ensuite scannées ou analysées par l'intermédiaire d'une application web qui mettra à jour la base de données.

Mots clés :

- Base de données
- MySQL
- web
- php

Sommaire :

I] Conception de la base de données.....	5
a) Le schéma relationnel.....	6
b) Les tables.....	6
II] Développement de la base de données et de l'application web....	8
a) Prérequis d'installation de logiciels.....	8
1) Installation de WAMP.....	8
2) Installation et utilisation d'une librairie GitHub.....	9
b) Mise en place de la structure de la base de données.....	9
c) Développement de l'application web.....	10
1) Organiser les fonctionnalités.....	10
2) Développement des fonctionnalités.....	12
• Ajout d'une liste d'étudiants à la BDD.....	12
• Ajout d'une liste de cours à la BDD.....	13
• Impression d'une fiche d'absences.....	16
• Scan/Analyse d'une fiche d'absences.....	17
3) Autres évolutions pour ce système.....	18
III] Utilisation et exemple d'application.....	19
IV] Gestion de projet.....	22

Introduction et problématique :

Polytech Annecy-Chambéry souhaite mettre en œuvre un système qui permet d'avoir un suivi et une supervision simple du système scolaire de l'école.

La supervision de ce système scolaire comprend la vision globale de tous les enseignants à disposition de l'école, des connaissances qui doivent être transmises, des cours planifiés, et surtout la présence des étudiants aux cours.

Actuellement, L'école ne possède pas un système informatisé qui assure une supervision totale de toute cette organisation. Ce manque de moyens implique une supervision moins efficace sur la réussite des étudiants, car leur présence aux cours n'est pas systématiquement vérifiée.

En effet, le personnel administratif prend déjà beaucoup de temps à effectuer manuellement le suivi des cours et des étudiants, à un point où ils ne peuvent pas se permettre de suivre tous les étudiants à chacun de leurs cours.

En particulier, ils conservent chaque absence relevée sous la forme de fichiers. L'étudiant concerné doit justifier cette absence plus tard, sur papier, pour que le personnel administratif mette à jour ce fichier.

Ainsi, le système à développer doit permettre de centraliser toutes les informations, d'avoir une supervision plus globale et plus accessible sur les données et surtout, permettre une mise à jour rapide de celles-ci sur la présence des étudiants aux cours.

Dans une première partie, nous décrirons la structure de la base de données nécessaire pour répondre au problème. Dans un deuxième temps, nous expliquerons comment nous avons mis en place cette base de données, ainsi que l'application qui interagira avec elle. Enfin, nous conclurons sur des exemples pratiques qui utiliseront le système développé, ainsi que sur la gestion de projet mise en place pour suivre la progression du travail.

I] Conception de la base de données

La base de données à mettre en œuvre doit permettre de représenter l'ensemble du schéma suivant :

« Un ensemble d'enseignants est présent pour instruire et transmettre les connaissances et compétences en ingénierie à un groupe d'étudiants, par l'intermédiaire de cours magistraux (CM), travaux dirigés (TD) et travaux pratiques (TP).

Différents modules existent pour assurer l'apprentissage de toutes les compétences nécessaires. Un cours est lié au programme d'un module, et chaque cours est dirigé par un enseignant.

Les étudiants sont classés selon leur filière, année de promotion, groupe de TD, et groupe de TP. Chaque catégorie d'étudiants doit suivre un ensemble de cours. »

Nous avons imaginé le schéma relationnel ci-dessous qui permet de représenter tous les éléments et leurs relations entre eux.

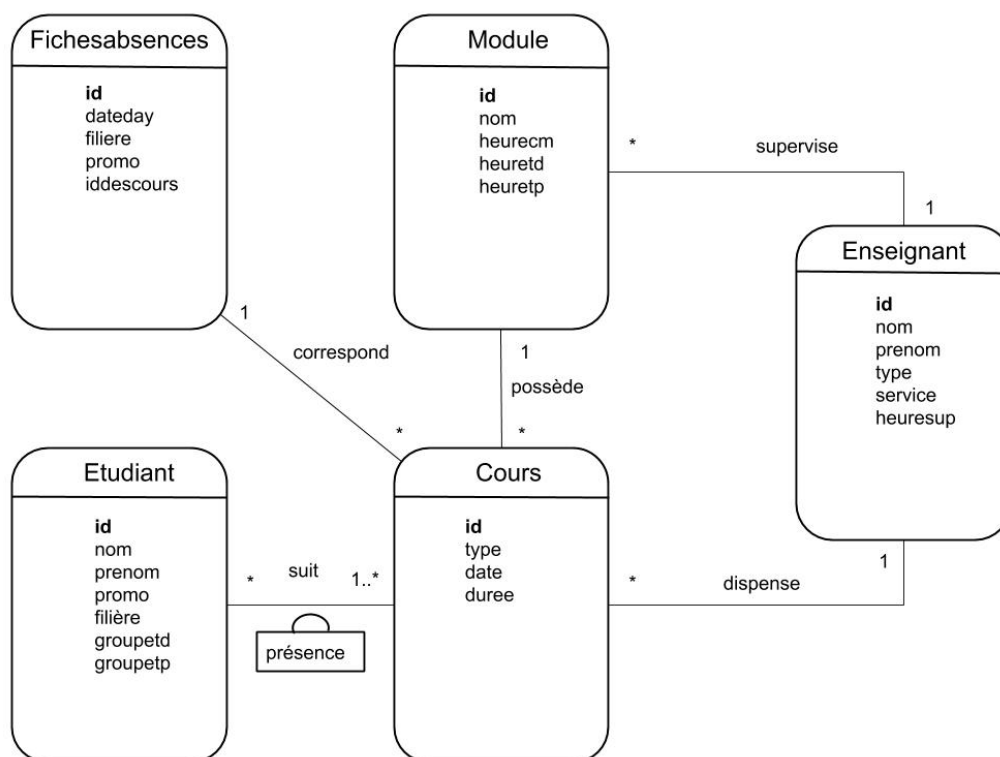


Figure 1: Schéma relationnel

a) Le schéma relationnel

Actuellement, l'école possède à sa disposition plusieurs enseignants. Ces derniers organisent les connaissances sous la forme de modules.

De cette façon, il est désigné un enseignant responsable pour chaque module qui doit organiser celui-ci. Il existe une relation d'un même enseignant à plusieurs modules (car un enseignant peut être responsable de plusieurs modules).

Pour les cours, il existe une relation qui relie plusieurs cours à un module, car plusieurs cours sont nécessaires pour transmettre les connaissances relatives à un module.

Il existe également une relation qui relie plusieurs cours à un enseignant, car un enseignant est amené à diriger plusieurs cours pour leur bon fonctionnement.

Entre les étudiants et les cours, c'est une relation de plusieurs étudiants à plusieurs cours qui existe. En effet, les étudiants sont organisés par groupes, et un même groupe doit assister à plusieurs cours.

Enfin, un cours ne figure que sur une seule fiche d'absences. Il existe donc une relation qui relie une même fiche d'absences à plusieurs cours.

b) Les tables

Dans une base de données, une table représente une entité qui va avoir des relations avec d'autres entités. Une entité est caractérisée par plusieurs attributs qui la définissent. Ces attributs permettent de distinguer deux entités différentes. Nous avons considéré que les attributs listés ci-dessous sont importants pour diverses raisons (expliqué pour chacun d'eux).

Ainsi, une table doit être créée pour représenter distinctement :

- Les enseignants : Ils ont un nom, un prénom, un nombre d'heures représentant leur durée de « service » en tant qu'enseignant. Ils sont également soit permanents soit vacataires, et sont autorisés ou non à effectuer des heures supplémentaires. Remplir ces informations est important afin de gérer plus facilement les services des différents enseignants.
- Les modules : Sous la responsabilité d'un enseignant, chaque module a un nom, et concerne un programme qui est enseigné au cours d'un semestre spécifique. Le programme consiste à enseigner un certain nombre d'heures de CM, de TD et de TP. Ces nombres d'heures sont essentiels afin d'avoir une meilleure gestion et un meilleur aperçu sur la façon dont les connaissances vont être transmises. L'enseignant responsable peut être « null », car il n'est pas toujours défini.
- Les étudiants : Chaque étudiant a un nom et un prénom. Il a une année de promotion qui marque la fin de ses études, mais qui désigne aussi indirectement le semestre d'étude qu'il est en train de poursuivre. Un étudiant a également une filière, un groupe de TD et un groupe de TP qui permettent de définir l'ensemble des cours auxquels il va devoir assister (avec l'année de promotion).

- Les cours : Un cours est soit un CM, soit un TD, soit un TP. Il permet d'enseigner des éléments correspondant à un module et il est dirigé par un enseignant. Il a une date marquant le jour et l'heure du début du cours. Le cours se déroule sur une certaine durée. Ces informations permettent de gérer plus facilement la répartition du temps et des compétences d'un module aux cours. Le module et l'enseignant auquel il est relié peuvent être « null » car ceux-ci ne sont pas toujours renseignés.

- Les présences : C'est la table correspondant à la relation de plusieurs étudiants à plusieurs cours. Elle permet de relier chaque étudiant à chaque cours avec leurs identifiants, ainsi que l'information si oui ou non, l'étudiant a été présent au cours.

- Les fiches d'absences : Nous avons choisi que le suivi de la présence des étudiants aux cours s'assurera par l'intermédiaire de fiches d'absences qu'il faudra imprimer, remplir, puis renseigner dans la base de données. Cette table représente toutes les fiches d'absences qui seront générées : Elle permet d'assurer l'association entre les étudiants et les cours lorsqu'il sera nécessaire de scanner ou d'analyser la fiche d'absences.

Nous avons également imaginé la possibilité d'insérer une sixième table au modèle, associée à la table étudiant, qui pourra enregistrer des justificatifs concernant l'absence d'un étudiant sur une période donnée (avec une date de début, date de fin, et la raison).

Le système développé ne permet pas d'enregistrer les justificatifs d'absences.

II] Développement de la base de données et de l'application web

A présent que nous avons conceptualisé la structure de la base de données dont nous avons besoin pour répondre à la problématique, il nous reste à implémenter cette base de données sur un serveur, et de développer une application qui permettra de se connecter à cette base de données pour la mettre à jour, ajouter des données, et recevoir toutes les informations nécessaires sur le suivi des cours.

Nous avons décidé de recourir à une application web afin d'interagir avec la base de données. En effet, nous souhaitons à terme que ce logiciel soit adaptable à chaque machine et que chaque utilisateur puisse accéder aux informations de la base de données et aux fonctionnalités de l'application qui les concerne.

Les logiciels et installations nécessaires utilisés pour le développement de la base de données vont être décrits, puis nous aborderons la conception et le développement de l'application web.

a) Prérequis d'installation de logiciels

Nous avons choisi d'utiliser WAMP (Windows, Apache, MySQL, PHP) afin de générer la base de données sur un serveur MySQL.

WAMP est couramment utilisé pour développer la base de données ainsi que les sites webs sur une même machine. Il unit toutes les bibliothèques nécessaires pour le bon fonctionnement de la base de données et du site web, ou de l'application web.

Il existe différents types de serveurs pour héberger une base de données, nous avons choisi MySQL car les bases de données de Polytech-Annecy utilisaient déjà des serveurs MySQL.

1) Installation de WAMP

Il faut installer WAMP sur la machine qui supportera la base de données.

Il est important de s'assurer au préalable d'avoir tous les prérequis nécessaires avant d'installer WAMP, auquel cas, le serveur web ne sera pas fonctionnel.

Il faut, en particulier, que toutes les versions de « Visual C++ » nécessaires pour l'utilisation de WAMP soient déjà installées sur l'ordinateur. Celles-ci sont renseignées dans un avertissement avant l'installation de WAMP. Il s'agit d'environnements de développement pour windows que WAMP réutilise.

2) Requis d'une librairie sur GitHub

Lors de ce projet, il sera nécessaire d'ajouter l'ensemble des étudiants à la base de données. Les informations dont nous avons besoin sur les étudiants sont renseignées dans plusieurs fichiers excels (.xls). L'extraction de ces informations n'est pas possible sans bibliothèque PHP extérieure. Je devais donc trouver un moyen d'implémentation pour que tous les élèves listés sur un fichier excel soient ajoutés à la base de données par l'intermédiaire d'une application. En outre, une librairie intitulée « PhpSpreadSheet » a été utilisée dans le but de lire et d'éditer des fichiers Excel avec php.

Cette librairie peut être installée en utilisant « composer », un outil qui permet d'importer plus facilement une bibliothèque complexe, tandis que la bibliothèque elle-même peut se trouver sur Git-hub.

b) Mise en place de la structure de la base de données

La base de données est accessible par l'intermédiaire de phpMyAdmin de WAMP. C'est un logiciel écrit en php, ce qui permet d'accéder à la base de données par le web.

Par défaut, la base de données est accessible avec « root » comme nom d'utilisateur et aucun mot de passe.

J'ai ensuite créé une nouvelle base de données, qui contiendra toutes les informations liées au projet, avec comme interclassement par défaut « latin1_swedish_ci ».

Puis, j'ai créé une nouvelle table pour chaque entité (enseignant, module, étudiant, cours, Présence et Justification), avec leurs attributs respectifs.

Nous utilisons comme moteur de stockage **InnoDB** qui est un moteur plus stable, plus récent et plus sécurisé lors de la manipulation des données.

Les clés étrangères représentant les relations d'un à plusieurs, entre deux tables, doivent être ajoutées manuellement avec la requête SQL adéquate.

La structure de la base de données peut être exportée sous la forme d'un fichier SQL. Ce fichier correspond au code SQL qui permet de générer la structure de la base de données. Il devient ainsi possible de redéployer facilement la structure sur un autre serveur.

c) Développement de l'application web

A présent, il faut développer une application web qui permette à n'importe quel utilisateur d'accéder facilement aux données, de les lire, de les modifier, et d'ajouter de nouvelles données.

1) Organiser les fonctionnalités

Les différentes données à ajouter, ou à mettre à jour sont les informations concernant :

- Les enseignants
- Les modules
- Les étudiants
- Les cours
- La présence relative de chaque étudiant à chacun de ses cours respectifs
- La justification d'absence d'un étudiant sur une période de temps limité.

Actuellement, l'école répartit les étudiants par promotions et par filières. On a donc accès à des fichiers excels (fichiers .xls) listant, par exemple, tous les étudiants de cinquième année d'étude en filière IAI.

Les cours sont répertoriés et organisés sur un serveur qui permet aux utilisateurs d'accéder à leur visibilité par l'intermédiaire d'ADE en ligne. Il est possible d'extraire tous les cours relatifs à une promotion d'étudiants, une filière d'étudiants, ou à un enseignant sur une période de temps définie, sous la forme d'un fichier iCal (fichier .ics).

Les enseignants, ainsi que les modules, sont ajoutés manuellement à la base de données. En effet, les modules peuvent être amenés à être modifiés individuellement entre deux années de scolarité, tandis que les enseignants peuvent entrer, changer de responsabilités, ou sortir de l'environnement de Polytech.

Enfin, la présence des étudiants à chaque cours doit être assurée : nous avons imaginé que l'application soit en mesure d'imprimer une fiche d'absences par filière par promotion d'étudiants pour la journée, et une fois la fiche remplie, elle est scannée ou analysée, pour que les informations soient automatiquement enregistrées dans la base de données.

En imaginant l'application, j'ai trouvé quatre cas d'utilisations :

- L'ajout d'une grande quantité d'étudiants ou de cours en une fois. Les étudiants sont tous ajoutés en début d'année scolaire tandis que les cours sont ajoutés en cours d'année.
- L'ajout ou la modification exceptionnels d'un élément dans la base de données (qui concerne un enseignant, un module, un étudiant, ou un cours).
- L'impression de la fiche d'absences des étudiants, pour une filière, promotion et journée donnée.
- Le scan ou l'analyse de la fiche d'absences remplie.

De cette manière, j'ai développé le menu principal afin que l'utilisateur puisse cliquer sur quatre liens menant chacun à l'un de ces cas d'utilisation.

Un cinquième cas d'utilisation partiellement développé est le cas où l'utilisateur souhaite consulter les absences enregistrées.

La mise en place des fonctionnalités a impliqué l'organisation du site web selon le schéma ci-dessous. Les encadrés en bleu représentent les fichiers .php qui vont ajouter ou modifier des éléments dans la base de données :

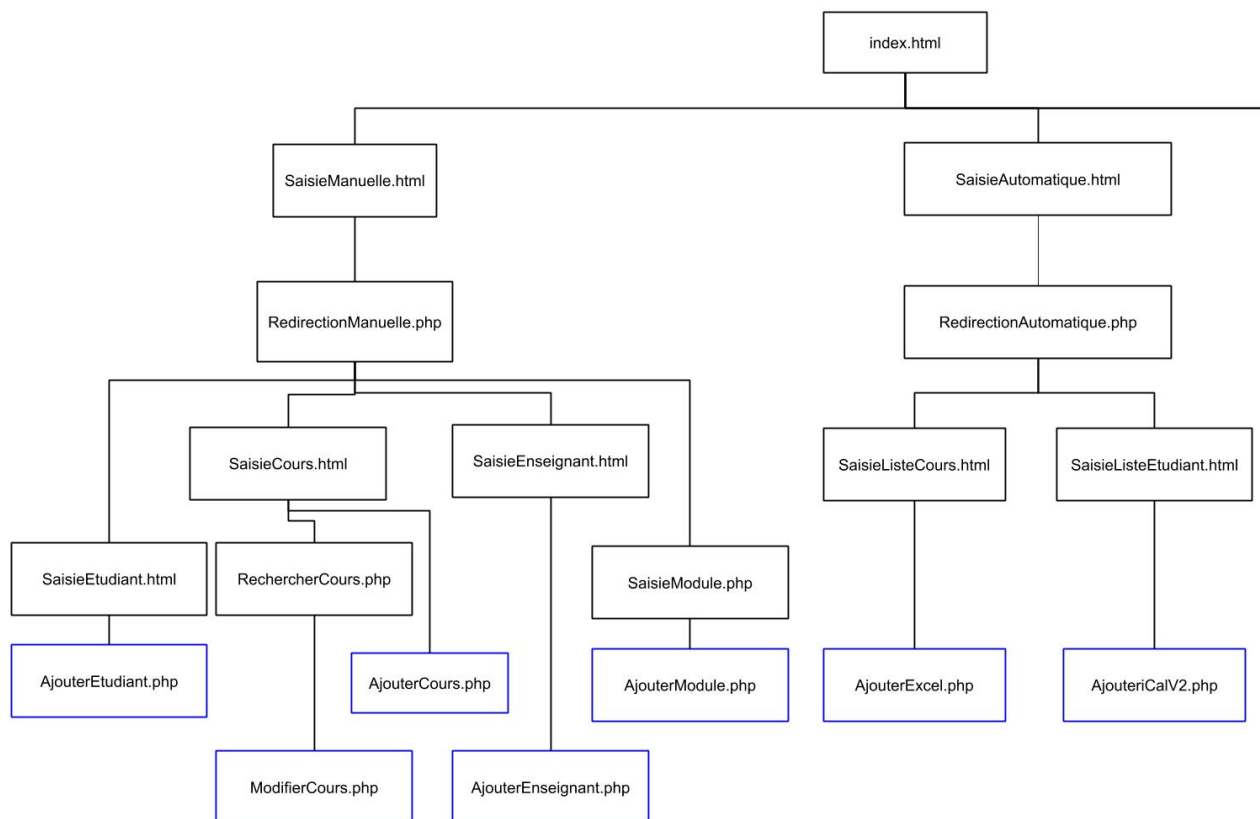


Figure 2 : Structure de l'application web 1/2

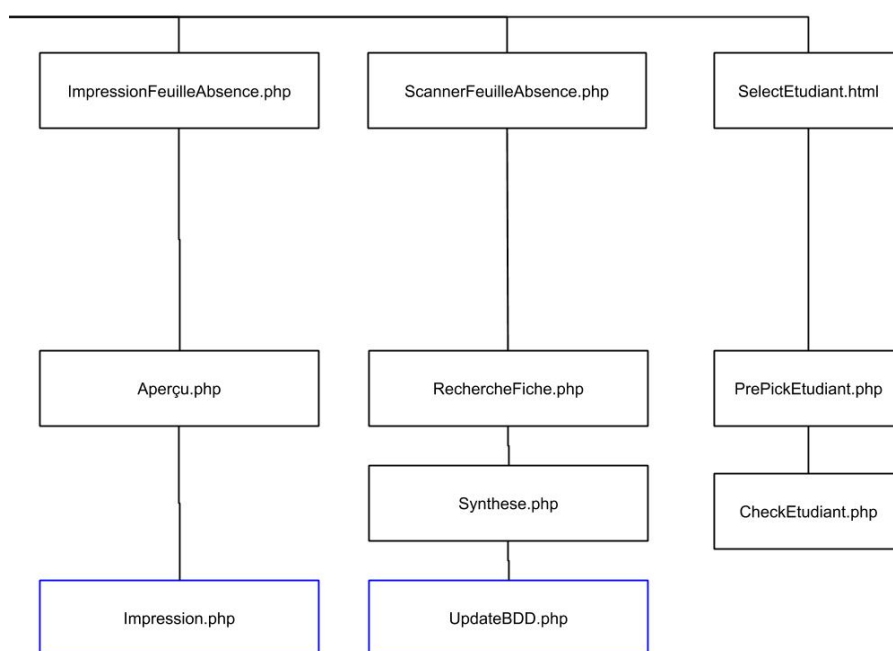


Figure 3 : Structure de l'application web 2/2

2) Développement des fonctionnalités

L'application web est organisée par plusieurs fichiers codés en html et php.

Je vais décrire le développement de chaque cas d'utilisation, les problèmes rencontrés, les problèmes qui persistent ou qui peuvent être source de problèmes à l'avenir, et les améliorations encore possibles à mettre en œuvre.

- **Connexion à la base de données et requêtes SQL :**

Il est nécessaire de réaliser une connexion entre l'application et la base de données afin d'obtenir certaines informations ou d'enrichir la base de données par le biais de requêtes SQL.

Pour réaliser cette connexion, j'ai choisi d'utiliser PDO.

Cela nous permet d'obtenir la réponse d'une requête SQL dans un seul tableau de valeurs.

Toutes les informations de cette requête sont ainsi transmises du serveur à l'application en une seule fois.

Autrement, les informations de connexions à la base de données peuvent être renseignées dans un fichier Config.php. Ce fichier assemble, en un seul endroit, tous les paramètres modifiables pour adapter l'application à l'utilisateur qui souhaiterait changer certains paramètres. Par exemple, le nom de la base de données ainsi que le port sont des informations pouvant être modifiées. Config.php comprend également d'autres paramètres utiles pour la bonne exécution des fonctionnalités de l'application.

Tous les autres fichiers PHP importent alors Config.php pour utiliser les mêmes paramètres.

- **Ajouter une liste d'étudiants à la base de données :**

Les étudiants à intégrer dans la base de données sont classés dans plusieurs fichiers Excels (.xls) différents, selon leur filière et année de promotion. Ces fichiers étaient utilisés par la scolarité pour classer les étudiants.

PHP n'a pas de bibliothèque ou de classes préinstallées pour lire ce type de fichier.

J'ai donc dû utiliser une bibliothèque extérieure nommée PhpSpreadsheet pour accéder aux informations du fichier dont nous avons besoin.

Sur le fichier Excel, on peut trouver toutes les informations nécessaires pour ajouter un étudiant par l'intermédiaire d'une requête SQL.

Grâce à PhpSpreadsheet, les informations contenues dans une cellule du fichier peuvent être récupérées dans une variable sous la forme d'un entier ou d'une chaîne de caractères.

Toutefois, l'information que je souhaitais extraire pouvait être associée à d'autres caractères ou informations que nous voulons distinguer ou ignorer.

Ainsi, la structure du fichier Excel initial a été modifiée pour que chaque information soit isolée dans une cellule distincte.

Par exemple, le nom du groupe de TP d'un étudiant était initialement contenu dans une cellule sous la chaîne de caractères « Groupe de TP D1 », donc j'ai isolé l'information importante, ici « D1 », dans une cellule unique, et la chaîne « Groupe de TP » dans une autre cellule.

Une fois toutes les informations obtenues pour un étudiant, nous pouvons l'ajouter dans la base de données avec une requête SQL.

Par ailleurs, il peut exister plusieurs «pages excel» dans un même fichier excel. Le programme développé permet de traiter distinctement chaque page excel (comme s'il s'agissait de fichiers excels différents).

Si la structure de ce fichier Excel est modifiée à l'avenir, les paramètres qui localisent les informations à extraire sont renseignés dans le fichier Config.php.

- **Ajouter une liste de cours à la base de données :**

Un ensemble de cours sur ADE peut être exporté sous la forme d'un fichier iCal.

Une simple classe PHP permet d'accéder aux informations contenues dans le fichier iCal.

Cette classe génère un tableau qui contient des « sous » tableaux pour chaque cours distinct.

Chacun de ces « sous » tableaux, relatif à un cours, détient toutes les informations dont nous avons besoin pour ajouter le cours à la base de données.

Toutefois, chaque information est décrite dans une chaîne de caractères qu'il faut exploiter au cas par cas. En effet, plusieurs informations importantes peuvent être présentes dans une seule chaîne de caractères, avec éventuellement des caractères indésirables. De plus il n'est pas possible de modifier la localisation des informations au moment d'exporter le fichier iCal.

La figure ci-dessous présente un « sous » tableau correspondant à un cours enregistré sur le fichier iCal

```
13 =>
    array (size=10)
      'DTSTAMP' => string '20200703T135336Z' (length=16)
      'DTSTART' => string '20191120T073000Z' (length=16)
      'DTEND' => string '20191120T090000Z' (length=16)
      'SUMMARY' => string 'SHES901_Management_CM' (length=21)
      'LOCATION' => string 'A-A204 (28 pl.)' (length=15)
      'DESCRIPTION' => string '
IAI-5
MMT-5
SCHRIVER MURIEL
(Exporté le:03/07/2020 15:53)
```

Figure 4 : Organisation des information dans un fichier iCal

Avec DTSTART et DTEND, nous pouvons obtenir la date de début du cours et déduire la durée de celui-ci avec la date de fin.

SUMMARY contient le nom du module déterminé par quatre lettres et trois chiffres, et parfois le type de cours (CM, TD ou TP), car il n'est pas toujours présent. Par exemple, tous les cours de LANG ne précisent pas le type de cours (un TD), ce cas particulier a été traité dans le code.

DESCRIPTION est la chaîne de caractères qui contient le plus d'informations, mais qui pose le plus de problèmes à exploiter. En effet, il contient chaque filière concernée, associée à l'année d'étude de cette filière, ainsi que le nom et prénom de l'enseignant qui a dirigé le cours.

De plus, il peut exister un certain nombre de caractères indésirables entre les informations, qu'il a fallu gérer pour extraire correctement les différentes informations de cette chaîne de caractères.

Le système actuellement développé permet d'extraire de la chaîne DESCRIPTION chaque filière associée à son année d'étude, ainsi que le nom et prénom de l'enseignant.

Toutefois, des problèmes peuvent survenir dans certaines conditions.

Afin de comprendre ces problèmes, il est nécessaire d'expliquer le raisonnement qui a mené au développement de cette solution.

Selon la façon dont la chaîne de caractères DESCRIPTION est construite, on constate les éléments suivants :

- Une lettre (d'alphabet) de la chaîne est dans la majorité des cas associée soit à un nom de filière, soit à un nom ou prénom d'un enseignant. Ainsi, pour extraire les informations de cette chaîne de caractères, j'ai choisi d'analyser un à un les éléments de la chaîne et de construire peu à peu une nouvelle chaîne de caractères avec les caractères de la chaîne DESCRIPTION, si le caractère correspond à une lettre. Si la chaîne construite s'avère être le nom d'une filière, d'un nom ou d'un prénom déjà présent dans la base de données, nous construisons une nouvelle chaîne.
- Un tiret succède toujours à un nom de filière et précède toujours l'année d'étude de cette filière. Donc si le caractère analysé est un tiret, nous vérifions si la chaîne de caractères construite correspond bien à un nom de filière existant. Si c'est le cas, le caractère suivant est un nombre correspondant à l'année d'étude.
- On trouve toujours un espace ou un retour à la ligne avant et après le prénom de l'enseignant.
- Dans la majorité des cas, les caractères de la chaîne sont tous collés entre eux (Dans l'exemple de la figure 4, La chaîne de caractères peut être enregistrée comme «IAI-5MMT-5SCHRIEVER MURIEL»).

Remarque notable : La filière « MM » est notée « MMT » dans le fichier iCal. Le système permet de traiter correctement ce nom de filière.

Le problème correspond aux autres éléments qui peuvent se trouver dans la chaîne DESCRIPTION.

En outre, plusieurs de ces éléments sont actuellement traités comme cas particuliers, notamment :

- La chaîne «EPU», qui correspond à toutes les filières.
- Une chaîne «-s5» ou «-s6» qui succède une année d'étude, qui correspond au semestre. Ce n'est pas une information utile, elle est ignorée.

Mais si d'autres cas particuliers existent, ils peuvent être source de problèmes, car non considérés comme cas particuliers.

En effet, les caractères qui suivent le caractère correspondant à l'année d'étude sont collés au nom de l'enseignant, ce qui pose problème lorsque ce nom doit être analysé.

Un élément d'amélioration pour le système actuel, est le traitement des chaînes de caractères «-TPA» et «-TPB» qui succède l'année d'étude.

C'est un élément important car il renseigne sur le groupe de TP concerné par le cours spécifié.

Toutefois, « TPA » et « TPB » ne correspondent pas nécessairement à des noms de groupe de TP présents dans la base de données. Nous ne connaissons pas non plus tous les noms possibles que l'on peut associer à un groupe de TP.

Notamment, dans les exemples de groupe d'étudiants que j'ai pu ajouter à la base de données, les noms des groupes de TP sont D1, D2, E1 et E2.

Nous n'avons ainsi pas la certitude que les caractères « TPA » désignent le groupe « D1 » ou « D2 ». Nous n'étions alors pas certains de pouvoir développer une solution qui prend correctement en compte les caractères « TPA » et « TPB », sachant que d'autres caractères peuvent exister pour désigner un groupe de TP.

Actuellement, les caractères « -TPA » et « -TPB » sont traités comme cas particuliers et sont ignorés.

Une fois toutes les informations sur le cours obtenues, nous réalisons une requête SQL qui ajoute le cours à la base de données.

Le module associé ainsi que l'enseignant ne sont pas toujours renseignés, et sont donc définis comme « NULL » s'ils sont manquants.

Il faut également ajouter des éléments dans la table « Présence » à l'ajout de chaque cours, afin de pouvoir enregistrer plus tard la présence des étudiants au cours.

Les informations sur la filière et l'année d'étude (et éventuellement, les groupes de TD et de TP) sont utilisées pour interroger la base de données et obtenir la liste de tous les étudiants concernés par le cours, et ainsi créer une relation entre le cours ajouté et chaque étudiant.

- **Ajouter un seul étudiant :**

L'application permet d'ajouter un étudiant à la base de données.

Elle ajoute également des éléments dans la table « Présence », qui permet d'associer le nouvel étudiant à chaque cours auquel il doit participer, selon l'année de promotion et la filière rentrée.

- **Ajouter un seul cours :**

L'application permet d'ajouter un cours à la base de données.

Toutefois, il n'y a pas encore de moyen mis en place pour préciser les étudiants concernés par ce cours, pour ajouter des éléments dans la table « présence ».

- **Ajouter un enseignant :**

L'application permet d'ajouter un enseignant à la base de données.

- **Ajouter un module**

L'application permet d'ajouter un module à la base de données.

Il est possible de ne pas l'associer immédiatement à un enseignant lors de la création du module.

L'enseignant peut être sélectionné parmi une liste construite à partir de tous les enseignants présents dans la base de données.

Un élément d'amélioration notable est de pouvoir modifier un étudiant, un module ou un enseignant, afin de corriger une faute ou de mettre à jour leurs données.

- **Modifier un cours**

L'application permet de modifier ou compléter un cours qui n'est pas encore associé à un module ou un enseignant.

L'utilisateur peut rentrer une date afin d'afficher tous les cours incomplets associés à cette date, ou aux dates qui contiennent la date rentrée. Par exemple, entrer seulement 2020 affichera tous les cours incomplets qui ont lieu en 2020, mais rentrer 2020-03 affichera tous les cours incomplets qui ont lieu en mars 2020.

Puis, les cours incomplets s'affichent et peuvent être mis à jour immédiatement : Des formulaires permettent à l'utilisateur de remplir le nom du module ou de l'enseignant manquant au cours.

Appuyer sur le bouton « modifier » applique les modifications, si les noms rentrés correspondent bien à des éléments déjà existant dans la base de données, et retire le cours de la liste.

Un élément d'amélioration serait d'ajouter également le prénom de l'enseignant, pour éviter une rare situation où deux enseignants portent le même nom (Il existe un cas à Polytech).

- **Impression d'une fiche d'absences**

Pour imprimer une fiche d'absences, je devais trouver un moyen de générer un document PDF structuré, contenant toutes les informations concernant la journée de cours d'une filière et promotion d'étudiants. Cela englobe les étudiants concernés et, pour chaque cours, la date de début, le type de cours, le module et l'enseignant concerné.

Un moyen pour générer un document PDF est d'utiliser PHPSpreadSheet, déjà utilisé dans le projet pour lire les fichiers .xls. Il propose des outils pour générer un document PDF.

Il nous est ainsi possible d'inscrire sur un « Spreadsheet » toutes les informations nécessaires, cellule par cellule, et de sauvegarder ce « Spreadsheet » sous un format PDF. Nous autorisons alors l'application Web à télécharger ce fichier PDF pour l'ouvrir et l'imprimer manuellement avec le logiciel qu'il a à sa disposition.

Toutefois, pour définir le style de la fiche d'absences, je pensais initialement copier le style d'un fichier modèle d'une fiche d'absences pour l'intégrer au Spreadsheet construit. Mais je ne parvenais pas à le faire avec PhpSpreadSheet.

Définir le style de la fiche d'absences peut se faire manuellement avec PhpSpreadSheet, en éditant chaque paramètre un à un, mais nécessite de trouver toutes les fonctions spécifiques et de les réutiliser pour obtenir une fiche d'absences similaire au modèle souhaité.

J'ai choisi d'intégrer un style minimal à la fiche d'absences générée pour que celle-ci soit fonctionnelle et lisible.

Autrement, il existe une bibliothèque « fpdf » pour générer un pdf manuellement. Il est probablement possible de générer une fiche d'absences mieux élaborée. De plus, « fpdf » supporte l'ajout d'images sur le pdf généré.

Une autre amélioration possible très notable, est de pouvoir afficher les cours dans l'ordre chronologique : Les premiers cours à gauche et les derniers cours à droite. Actuellement, les cours sont affichés dans l'ordre des identifiants des cours, ce qui n'est pas forcément l'ordre chronologique.

- **Scan ou analyse d'une fiche d'absences**

Un dernier problème à résoudre, qui concerne également l'impression, consiste à relier les informations complétées de la fiche imprimée (quel étudiant est absent) aux cours concernés. En effet, l'utilisateur ou la machine qui devra scanner ou analyser la fiche d'absences, doit pouvoir associer les identifiants des cours à celui des étudiants de la façon la plus simple possible.

J'ai alors décidé d'utiliser une table dans la base de données « fichesabsences » qui permet d'enregistrer la filière, la promotion, et la liste des identifiants des cours dans la base de données, ce qui représente une fiche d'absences. Cette solution permet de générer un identifiant unique pour chaque feuille d'absences que nous pourrons ajouter dans le coin supérieur gauche de la fiche avant l'impression. Ainsi, en scannant ou analysant cet identifiant de fiche, nous pourrons connaître l'ensemble des étudiants concernés ainsi que les cours, afin de compléter correctement la table présence lors de l'analyse de la fiche d'absences complétée.

Concernant le scan en lui-même, nous avons deux choix possibles.

Le premier consiste à automatiser au mieux ce procédé, en scannant la fiche à partir de l'application web, charger l'image sur le serveur, réaliser une analyse d'image pour détecter les étudiants absents aux cours, et mettre à jour la base de données en conséquence.

Le deuxième choix correspond à une approche plus simple et consiste à remplir un formulaire html, généré selon l'identifiant de la fiche d'absences entré, pour que l'utilisateur coche les étudiants absents aux cours.

Pour obtenir un système fonctionnel à temps, j'ai développé la deuxième solution.

Bien que la première solution soit réalisable, il est actuellement complexe d'utiliser un scanner à partir de l'application web, aucune bibliothèque PHP ne permettait de le faire, tandis qu'il est nécessaire de charger et de prendre connaissance d'une bibliothèque PHP en mesure de réaliser des analyses d'image.

Un des problèmes à résoudre était de transmettre correctement l'information selon l'étudiant absent à un cours donné.

Pour y résoudre, le formulaire propose une case à cocher pour chaque étudiant et chaque cours, ce qui résulte un grand nombre de cases à cocher, mais l'information sur la case à cocher n'est transmise par la méthode POST que lorsque cette case est cochée, tandis que l'information de cette case correspond à l'identifiant de l'étudiant avec l'identifiant de ce cours.

Nous pouvons alors d'abord réaliser les requêtes SQL pour informer les étudiants absents au cours d'une part, pour ensuite réaliser les requêtes SQL sur tous les autres étudiants à tous les cours concernés d'une autre part, pour préciser qu'ils ont bien été présents aux cours.

J'ai décrit toutes les fonctionnalités de notre application.

3) Autres évolutions pour ce système

Je vais résumer les évolutions possibles majeures déjà mentionnées du système actuel :

- Ajouter une meilleure maniabilité sur les données déjà ajoutées à la base de données.

En effet, le système ne permet pas de modifier un étudiant, un enseignant, un module ou la date d'un cours. Il est toujours possible de supprimer l'étudiant, l'enseignant ou le module pour en créer un nouveau, mais cette procédure n'est possible qu'avant le début d'une nouvelle année d'étude.

- Le style de la fiche d'absences peut être nettement amélioré pour correspondre au modèle original.

- Le scan automatique de la fiche d'absences et la mise à jour de la base de données peuvent être entièrement automatisés.

Outre ces évolutions possibles, d'autres fonctionnalités sont réalisables pour améliorer le système.

Un menu permettant de consulter les absences relatives à un étudiant a été partiellement développé. Il peut être amélioré afin d'apporter des informations plus précises sur les cours manqués.

L'application pourrait proposer à l'utilisateur de se connecter avec un login et mot de passe pour accéder aux fonctionnalités qui ne concernent que lui.

Ainsi, les étudiants pourraient se connecter à l'application pour, par exemple, consulter les cours qu'ils ont manqués ou justifier leur absence (non implémentés actuellement). Ou encore, les enseignants n'auraient accès qu'à la gestion manuelle des cours et des modules, tandis que la scolarité accéderait à la plupart des fonctionnalités.

III] Utilisation et exemple d'application

A présent, je vais mener une démonstration des fonctionnalités majeures de l'application.

- Ajout d'une liste d'étudiants :

La page SaisieListeEtudiants.html permet de sélectionner le fichier ci-dessous (il peut être de différents formats), il appelle la page AjouterExcel.php pour ajouter la liste des étudiants à la base de données :

	A	B	C	E	F	G	H	I
1			Spécialité	IAI		5 ^{ème} année		
2						Année 2019-2020		
3	Enseignant :		Module :	TP TD ou CM		Date :		
4								
5	Groupe de TP	Red : modules obligatoires	NOM	PRENOM			Groupe de TD	
6								
7		1	BERTON	Nicolas			1	
8		2	CARRY	Guillaume			2	
9		3	COVES	Dorian			3	
10		4	DA SILVA RODRIGUES	Hugo			4	
11		5	BOISSEAU	Nicolas			5	
12		6	EXCOFFIER	Clément			6	
13		7	MATH041, PHYS041, LANG001	GRUNER	Théo		7	
14		8	KRULIC	Nathan			8	
15		9	LANG	Nicolas			9	
16		10	LAVOREL	Tanguy			10	
17		11	MICHALLET	Denis			11	

Toutes les informations importantes sont situées sur une cellule unique.

Figure 5 : Structure du fichier avec une liste d'étudiants

	idetudiant	nom	prenom	promo	filiere	groupe td	groupe tp
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	BERTON	Nicolas	2020	IAI	D	D1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	CARRY	Guillaume	2020	IAI	D	D1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	COVES	Dorian	2020	IAI	D	D1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	DA SILVA RODRIGUES	Hugo	2020	IAI	D	D1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	BOISSEAU	Nicolas	2020	IAI	D	D1
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	EXCOFFIER	Clément	2020	IAI	D	D1

Figure 6 : Visualisation de la table Etudiant après lecture du fichier

- Ajout d'un ensemble de cours :

La page SaisieListeCours.html permet de sélectionner un fichier iCal.

Il appelle la page AjouterCalV2.php pour ajouter l'ensemble des cours à la base de données.

Ci-dessous, la représentation de deux cours dans le fichier iCal, ainsi que des enseignants et modules déjà présents dans la base de données :

```

1 =>
  array (size=10)
    'DTSTAMP' => string '20200703T135336Z' (length=16)
    'DTSTART' => string '20191122T123000Z' (length=16)
    'DTEND' => string '20191122T140000Z' (length=16)
    'SUMMARY' => string 'EASI943_CM' (length=10)
    'LOCATION' => string 'A-C103 (30 pl.)' (length=15)
    'DESCRIPTION' => string ''

IAI-5
MOUILLE PASCAL
(Exporté le:03/07/2020 15:53)

```

Figure 7 : Représentation d'un cours dans le fichier iCal

Mouille Pascal est un enseignant déjà présent dans la base de données.

Ainsi, lors de l'ajout du cours, celui-ci va être relié à l'identifiant de l'enseignant. Il en est de même pour le nom du module.

```

11 =>
  array (size=10)
    'DTSTAMP' => string '20200703T135336Z' (length=16)
    'DTSTART' => string '20191118T070000Z' (length=16)
    'DTEND' => string '20191118T110000Z' (length=16)
    'SUMMARY' => string 'INFO942_TPA' (length=11)
    'LOCATION' => string 'A-C209 (28 pl.)' (length=15)
    'DESCRIPTION' => string ''

IAI-5-TPA
ATTO ABDOURRAHMANE
(Exporté le:03/07/2020 15:53)

```

Figure 8 : Représentation d'un deuxième cours dans le fichier iCal

Si le module ou l'enseignant n'est pas présent dans la base de données, l'identifiant de celui-ci reste NULL.

	idenseignant	nom	prenom	type	service	heuresup
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	Vernier	Flavien	permanant	200	oui
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Mouille	Pascal	permanant	120	non
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	Badel	Adrien	permanant	200	oui
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	SCHRIVER	Muriel	vacataire	50	non

Figure 9 : Enseignants déjà présents dans la base de données

	idcours	type	datecours	duree	idmodule	idenseignant
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	TP	2019-11-21 08:00:00	8	NULL	NULL
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	CM	2019-11-22 12:30:00	1.5	1	2
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	11	TD	2019-11-18 17:00:00	1.75	7	NULL

Figure 10 : Résultat obtenu en ajoutant les cours

L'ajout du cours a aussi ajouté des éléments dans la table présence, selon les étudiants concernés. Pour cet exemple, les étudiants en IAI 5ème année sont bien concernés par les cours ajoutés.

idetudiant	idcours	presence	justificatif
1	1	NULL	non
2	1	NULL	non
3	1	NULL	non
4	1	NULL	non

Figure 11: Elements ajoutés dans la table Présence

- Imprimer une fiche d'absences :

J'imprime la fiche d'absences en sélectionnant d'abord le jour, et la filière/année d'étude des étudiants concernés.

Je sélectionne le 22 novembre, jour où le cours à l'identifiant id=2, cours de EASI943 avec Mr MOUILLE a lieu à 12h30.

Jour : 22 Novembre 2019 | Classe : IAI 5
12h30 - 14h0 | Module : EASI943 | Enseignant : Mouille
14h15 - 15h45 | Module : EASI943 | Enseignant : Mouille
07h00 - 11h0 | Module : EASI941 | Enseignant : Badel

Tous les cours pour la fiche absence ont été affichés
 Veuillez patienter lors de la génération du PDF

J'obtiens bien la présence de ce cours, avec deux autres cours durant la même journée.

Figure 12 : Prévisualisation des cours de la journée avant l'impression

La fiche d'absences complète peut être trouvée en **annexe 1**. J'y trouve l'identifiant de la feuille (ici, 1) dans le coin supérieur gauche de la fiche.

- Analyse de la fiche d'absences :

Une fois complétée, la fiche d'absences peut être analysée par une personne qui va utiliser l'application pour remplir un formulaire composé de cases à cocher, pour cocher les étudiants absents aux cours.

Après avoir sélectionné le jour concerné, puis l'ID de la fiche d'absences à considérer, le formulaire adéquat apparaît.

Je suppose que le premier étudiant (ID = 1) est absent aux deux premiers cours, le deuxième étudiant (ID=2) au troisième cours, et le troisième étudiant (ID = 3) à tous les cours de la journée. Les id des cours correspondent aux ID 2, 5 et 9.

Absent au cours 1 ☒ au cours 2 ☒ au cours 3 ☐ Absent à tous les cours de la journée ☐ BERTON Nicolas
 Absent au cours 1 ☐ au cours 2 ☐ au cours 3 ☒ Absent à tous les cours de la journée ☐ CARRY Guillaume
 Absent au cours 1 ☐ au cours 2 ☐ au cours 3 ☐ Absent à tous les cours de la journée ☒ COVES Dorian
 Absent au cours 1 ☐ au cours 2 ☐ au cours 3 ☐ Absent à tous les cours de la journée ☐ DA SILVA RODRIGUES Hugo

Figure 13 : Visualisation d'un formulaire rempli

idetudiant	idcours	presence	justificatif	idetudiant	idcours	presence	justificatif
1	2	absent	non	1	9	present	non
2	2	present	non	2	9	absent	non
3	2	absent	non	3	9	absent	non
4	2	present	non	4	9	present	non

idetudiant	idcours	presence	justificatif
1	5	absent	non
2	5	present	non
3	5	absent	non
4	5	present	non

Ainsi, la base de données a correctement été mise à jour selon les informations remplies sur le formulaire.

Figure 14 : Base de donnée présence mise à jour

IV] Gestion de projet

Pour mener à bien ce projet, nous avons utilisé divers outils pour suivre ou conserver la progression du travail.

Pour suivre la progression du travail, j'ai utilisé des diagrammes de Gantt pour planifier les tâches, les organiser et mettre à jour leur progression chaque semaine, afin d'estimer le travail qu'il reste à réaliser.

En annexe 2 et 3, nous pouvons trouver des diagrammes de Gantt correspondant aux diagrammes de début et de fin de projet.

Des réunions étaient réalisées chaque semaine avec le tuteur en visioconférence pour suivre l'avancée du projet, discuter des problèmes rencontrés, des solutions à envisager et des problèmes à résoudre.

Chaque semaine, avant chacune de ces réunions, je réalisais un rapport de la semaine pour faire le point sur l'avancée du travail, les problèmes rencontrés, et les tâches qui restaient à réaliser la semaine suivante.

Enfin, pour conserver le travail réalisé, nous avons utilisé Git-Hub pour déposer fréquemment, au moins chaque semaine, le code du projet que j'ai développé, afin d'en garder une trace en cas de problème majeur.

Conclusion

Le projet nécessitait de développer un système permettant une supervision simple du fonctionnement de l'école. Celui-ci devait également suivre de façon totale la présence des étudiants aux cours n'étant initialement pas totalement assurée.

Le système développé et proposé permet d'avoir une supervision sur tous les enseignants, modules, cours, étudiants, ainsi que leur présence aux cours, avec une base de données. Chaque élément de cette base de données peut être ajouté à l'aide d'une application web. Cette application peut générer et imprimer une fiche d'absences quotidienne dédiée à un groupe d'étudiants. Il propose à l'utilisateur de remplir un formulaire pour enregistrer la présence des étudiants aux cours.

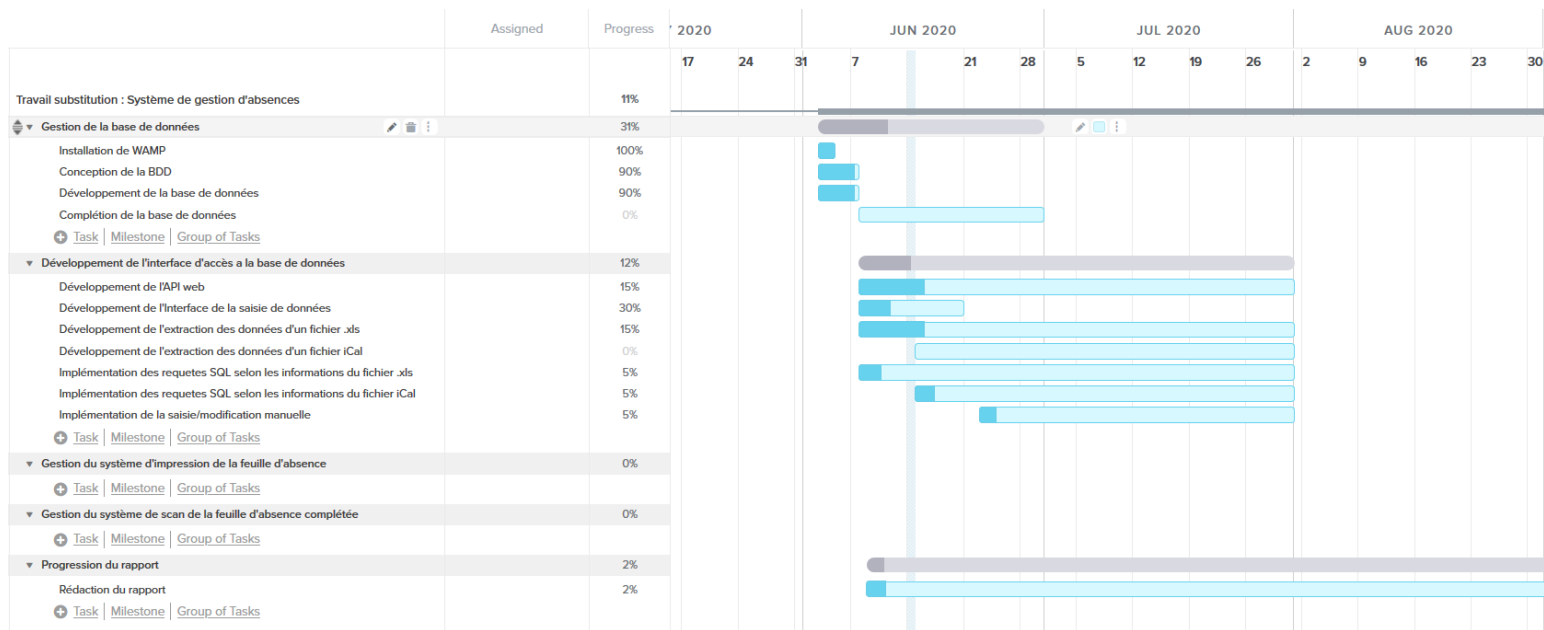
Dans un second temps, nous pourrions améliorer le système en intégrant un scan complètement automatisé de la fiche d'absences, en perfectionnant le rendu de la fiche d'absences et en ajoutant d'autres fonctionnalités utiles.

Annexe

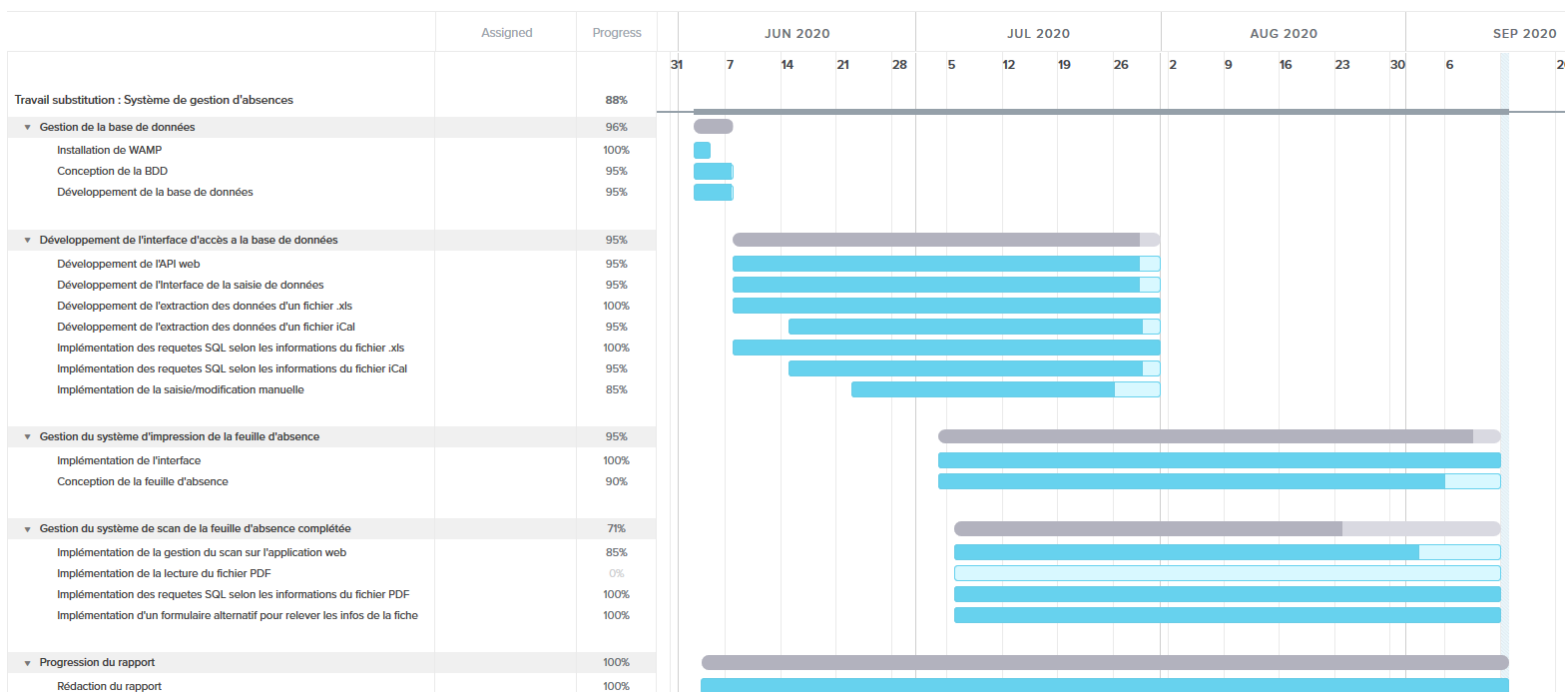
Annexe 1 : Exemple d'une fiche d'absences

ID : 26	Date :	2019-11-20	IAI-5 12h00 TP signature	IAI-5 SHES901 09h15 CM SCHRIVERsignature	IAI-5 11h00 CM signature	IAI-5 LANG70217h00 TD signature	IAI-5 SHES901 07h30 CM SCHRIVERsignature
NOM	Prénom	N/A	N/A	N/A	N/A	N/A	N/A
BERTON	Nicolas						
CARRY	Guillaume						
COVES	Dorian						
DA SILVA							
RODRIGUES	Hugo						
BOISSEAU	Nicolas						
EXCOFFIER	Clément						
GRUNER	Théo						
KRULIC	Nathan						
LANG	Nicolas						
LAVOREL	Tanguy						
MICHALLET	Denis						
PALAU	Clément						
PROTO	Geoffrey						
SEGUIN	Valentin						
DETIEGE	Tom						
DELATTRE	Nicolas						
DUCROT	Maxime						
HOUHA	Ryan						
MANTONNIER	Ulysse						
MICHEL	Nicolas						

Annexe 2 : Diagramme de Gantt du 15 Juin (début de projet)



Annexe 3 : Diagramme de Gantt du 13 septembre (Fin de projet)



Références :

- Bibliothèque PhpSpreadsheet :

<https://github.com/PHPOffice/PhpSpreadsheet>