

# Projet

## Gestion électronique des relevés de notes

### Cadre

L'objectif du projet est le développement d'une application Web/PHP/MySQL à l'aide du framework vu en cours de Web. Le travail se fait seul ou en binôme au cours des semaines 01 et 02. Il fera l'objet d'une démonstration à l'issue des travaux.

### Introduction

Le but de ce projet est la réalisation d'une application Web de gestion électronique des relevés de notes. Actuellement à 3iL, lorsqu'une épreuve est donnée aux élèves, un relevé de note électronique (fichier Excel) est édité par la Vie Etudiante puis envoyé par mail à l'enseignant qui le remplit une fois l'épreuve corrigée. Par épreuve on entend EI, DS, TP Noté, Projet etc.

Sur ce relevé figure la liste des élèves qui ont participé à l'épreuve, chacun devant avoir une note où être désigné absent.

Une fois rempli par l'enseignant, le relevé est retourné à la Vie Etudiante qui importe les données dans les bases de l'école.

En théorie, un enseignant a 3 semaines pour rendre les notes. Ce délai peut être rallongé ou raccourci suivant le contexte (proximité d'un jury).

Ce traitement est encore grandement manuel présente plusieurs défauts :

- Il n'existe aucun tableau de bord électronique, ni pour la Vie Etudiante, ni pour les enseignants pour savoir quels sont les relevés faits / en attente / en retard.
- Les échanges par mails sont source de perte de documents, d'oubli de pièce jointe, d'erreur d'adresse.
- Rien ne permet d'observer le respect des délais.
- Gestion chronophage.

### Périmètre

Le périmètre du projet concerne la création d'un outil permettant de gérer l'ensemble de cette activité.

- Pour la vie étudiante :
  - o Créer des relevés de notes, les assigner à un ou plusieurs enseignants.
  - o Réceptionner des relevés de notes.
  - o Marquer l'état de traitement d'un relevé de note
  - o Avoir une vision de ce qui est fait / en cours / en retard.
  - o Pouvoir répondre rapidement aux élèves qui demandent si certaines notes ont été rendues.
- Pour un enseignant :
  - o Recevoir, remplir et transmettre des relevés de notes
  - o Avoir une vision des relevés de notes en attente

Sont hors-périmètre :

- La gestion de l'importation des notes dans les bases de l'école (c'est une base spécifique).
- La gestion des groupes d'élèves (cycle, promotion, groupe de TD)
- Tout calcul nécessitant plusieurs relevés de notes (exemple moyenne d'un élève).

## Utilisateurs de l'application

Les types d'utilisateurs sont :

- Un administrateur qui gère les utilisateurs.
- Le personnel de la Vie Etudiante qui gère les relevés.
- Les enseignants qui remplissent les relevés.

A chacun de ces types correspond une interface différente et adaptée. Un compte n'appartient qu'à une seule catégorie et n'accède donc qu'à une seule interface. Il n'existe qu'une seule et unique page de connexion. L'utilisateur doit automatiquement être dirigé vers l'interface qui le concerne.

## Interface Vie Etudiante

L'idée est de disposer d'une interface "façon mail" pour gérer les relevés. Une colonne à gauche de l'écran présentera les catégories ci-dessous et à droite un datagrid dont le contenu change en fonction de la catégorie choisie. Les colonnes du datagrid ne sont pas forcément les mêmes pour chaque catégorie. A droite des noms des catégories, un nombre indiquera la quantité de relevés présents.

- Nouveaux : les relevés rendus par les enseignants mais pas encore traités.
- Brouillons : les relevés en cours de préparation mais pas encore envoyés.
- En attente : les relevés envoyés aux enseignants mais pas encore rendu.
- Traités : les relevés rendus par les enseignants et dont les notes ont été reportées dans la base de notes de l'école.
- Promotions : c'est un dossier qui regroupe les catégories correspondant à chaque promotion (CPI1, CPI2, I1, I2, I3, B3, M1, M2). Ce sont des raccourcis/filtres pour accéder aux relevés propres à une promotion, qu'ils soient "Nouveaux", "En attente" ou "Traités".

Quelques remarques :

- Les colonnes des datagrid doivent pouvoir servir de critère de tri.
- Dans la catégorie en attente une indication par un code de couleur permettra de distinguer rapidement les relevés en retard (rouge), à moins d'une semaine de la date prévue (jaune). Une simple pastille de couleur peut jouer ce rôle.
- Les colonnes doivent donner les informations les plus importantes.
- La Vie Etudiante doit pouvoir se substituer à un enseignant pour le remplissage d'un relevé (absence d'un enseignant, rendu papier ou mail de notes).

## Données d'un relevé

Les données d'un relevé sont les suivantes :

- Le cursus (3iL, Master, Bachelor)
- L'année scolaire (2013-2014)
- Matière
- Promotion (I1, I2, I3, B3, etc.)

- Date de l'épreuve
- Le ou les enseignants à qui sera transmis le relevé
- Le type d'épreuve (DI, DS, TP, Mini-Projet, etc.)
- Date de retour prévue du relevé.
- La liste des élèves concernés par l'épreuve.

Les colonnes de la liste des élèves sont :

- Le numéro de la ligne
- Le nom et prénom
- FE/FA
- La note
- L'observation de la Vie Etudiante (rattrapage, observation lors de l'épreuve, etc.)
- L'observation de l'enseignant (rien rendu, copie, etc.)

Remarques :

- Lorsqu'un relevé est créé par la Vie Etudiante, l'absence d'un élève à une épreuve est indiqué par la mention ABS dans la colonne note.
- Le relevé doit pouvoir être créé en plusieurs étapes avant d'être envoyé aux enseignants.

L'ergonomie de la création d'un relevé est importante. De prime abord on pourrait penser cela simple. Par exemple pour un DS concernant les B3, il suffit de choisir « B3 » pour ajouter tous les élèves de B3. Sauf que, cette année est un parfait contre-exemple, puisque jusque mi-janvier il faut ajouter les 4 élèves de I1. Peuvent aussi se rajouter des élèves de l'année dernière (possiblement actuellement en M1) et qui devraient passer un rattrapage.

De même, les rattrapages de session 2, pouvant ne concerner qu'une minorité d'élèves, de 1 à n.

De ce fait, on doit pouvoir facilement pouvoir ajouter ET retirer une promotion, un groupe ou des élèves individuellement. Il est aussi souhaitable de pouvoir disposer de combobox pour bénéficier de la complétion d'un nom d'élève.

## Interface enseignant

L'interface enseignant doit aussi reprendre le design "façon mail" avec une liste de catégorie à gauche et à droite le datagrid. Un enseignant ne voit que les relevés qui le concernent.

Les catégories demandées sont :

- Nouveaux : les relevés non remplis
- En cours : les relevés partiellement rempli ou non encore envoyés
- Envoyés : la liste des relevés remplis et envoyés.
- Promotions : la liste de tous les relevés groupés par promotion (sur le même principe que pour l'interface de la Vie Etudiante).

Lors du remplissage d'un relevé, l'ergonomie est aussi très importante. En effet, lors d'une correction les copies ou TP ne sont pas triés, les notes ne sont donc pas rentrées dans l'ordre alphabétique. Les listes pouvant être longue il serait souhaitable d'éviter les allées et retour clavier – souris. Exemple :

- Lorsqu'on est dans la zone de saisie d'une note, appuyer sur TAB conduit à la zone de saisie du commentaire, en appuyant sur TAB on passe à la zone de saisie de note de la ligne suivante. Une pression Majuscule-TAB permet d'aller vers le précédent. Techniquement les champs d'un formulaire ont un attribut tabindex qui permet de préciser l'ordre des éléments pour naviguer avec la touche TAB.
- Lorsqu'on est dans la zone de saisie d'une note, la pression sur une lettre permet de se déplacer sur la zone de saisie d'une note de l'élève dont le nom commence par soit la lettre tapée soit la lettre la plus proche.

## Points divers

- On prendra soin d'horodater les manipulations des relevés avec date de création ainsi que la date de modification séparée.
- Les enseignants et les types d'épreuves sont en base de données, il n'est pas de demandé d'interface pour les gérer.
- Pour gérer l'appartenance aux différentes catégories, on peut utiliser un ou plusieurs attributs "statut" avec un code.

## Contraintes techniques

- **IMPERATIF** : reprendre l'architecture MVC d'une application Web vue en TP. La partie framework peut être réutilisée telle qu'elle et enrichie.
- **IMPERATIF** : utilisation de Bootstrap Twitter pour le design de l'interface.
- **FACULTATIF** : utilisation de jQuery
- **RECOMMANDÉ** : utilisation d'une bibliothèque pour mettre en place les datagrids.
- **FACULTATIF** : utilisation d'autres bibliothèques
- **FACULTATIF** : utilisation de HTML5 / CSS3

## Déroulement du mini-projet

Lancement : semaine 46

Soutenance : semaine 50

Libre-service : 14 séances

- Les groupes formés lors de la séance de lancement ne doivent pas changer en cours de route sans validation de l'enseignant. Les élèves absents lors de cette séance seront soit prévu dans un groupe, soit seront ajouté à un groupe existant désigné par l'enseignant.
- L'horaire de passage de la soutenance est déterminé lors de la séance de présentation.
- La soutenance doit durer environ 20 à 30 minutes, questions comprises.
- L'organisation du travail est laissée libre aux élèves. Le MVC facilite le découpage et la spécialisation des rôles.
- La soutenance est une démonstration laissée libre aux élèves. **TOUS** les élèves du groupe doivent présenter ce sur quoi ils ont travaillé. A tout moment les enseignants peuvent poser de questions de tout ordre relative au projet, y compris sur le code du projet. Le code n'est pas à rendre, mais peut faire l'objet de vérification.
- La soutenance a lieu en salle de TP.
- Les démonstrations peuvent se faire sur les machines personnelles des élèves.
- En cas de manquement important de la part d'un élève (absences répétée aux séances, contribution quasi nulle au projet), les élèves du groupe devront en faire part aux enseignants.
- En cas de participation jugée insuffisante lors de la soutenance, les enseignants peuvent décider de dissocier la note de l'élève fautif.
- Aucun code n'est à rendre, toutefois il peut être demandé.
- La présentation peut être agrémentée de document pouvant faciliter la compréhension de ce qui a été fait (schéma de la BD, etc.). Mais il n'y a pas de Powerpoint à préparer.
- Le design de l'interface et l'ergonomie sont importants.
- Des fonctionnalités non précisées dans le sujet peuvent être ajoutée, **MAIS** seulement lorsque toutes les fonctionnalités demandées ont été incluses.
- L'idée est de montrer à la fois que vous avez bien répondu au sujet et que techniquement ce que vous avez fait tient la route.