

Projet GM4
MATHYS Dayane
MIGOT Tangi

Notice utilisation : Projet Visualisation des données

Sommaire

Configuration :

- on ne démarre pas juste une page web mais une application, configuration du fichier hosts.
- Logiciel utilisé : WampServer // phpMyAdmin

Base de données :

- Lien avec la base de données dans le code
- organisation des tables dans la base de données

Diagrammes d'utilisation :

- diagrammes séquences
- diagrammes des cas d'utilisations

Organisation du modèle et des applications :

- diagrammes des classes de la bibliothèque
- diagrammes des classes des applications
- composants web
- gestion des données

Utilisation du site Web :

- Utilisation Frontend
- Utilisation Backend

Configuration :

Les tests du site internet ont été effectués avec le serveur web WampServer, qui permet d'exécuter le code PHP en local.

Dans le dossier wamp/www il suffit de placer le site internet qui sera accessible à partir du serveur local. Afin qu'au démarrage du site, ça soit bien une application qui soit lancée (autrement wamp va demander quelle page doit-il exécuter en premier), on modifie le fichier « hosts » présent dans les fichiers de configuration de la machine, pour y insérer le code suivant :

```
#      127.0.0.1      ProjetPHP
```

Cette ligne spécifie qu'au démarrage du serveur local, on se dirige directement vers le site « ProjetPHP ». Pour que ça soit l'application qui démarre en premier on modifie un fichier de configuration du serveur Apache : Apache/httpd.conf

On ajoute le code suivant :

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@localhost
```

```
# Mettez ici le nom de domaine que vous avez utilisé dans le fichier hosts.
```

```
ServerName ProjetPHP
```

```
# Mettez ici le chemin vers lequel doit pointer le domaine.
```

```
DocumentRoot C:\Wamp\www\ProjetPHP\Web
```

```
<Directory C:\Wamp\www\ProjetPHP\Web>
```

```
Options Indexes FollowSymLinks MultiViews
```

```
# Cette directive permet d'activer les .htaccess.
```

```
AllowOverride All
```

```
# Si le serveur est accessible via l'Internet mais que vous n'en faites qu'une utilisation  
personnelle
```

```
# pensez à interdire l'accès à tout le monde
```

```
# sauf au localhost, sinon vous ne pourrez pas y accéder !
```

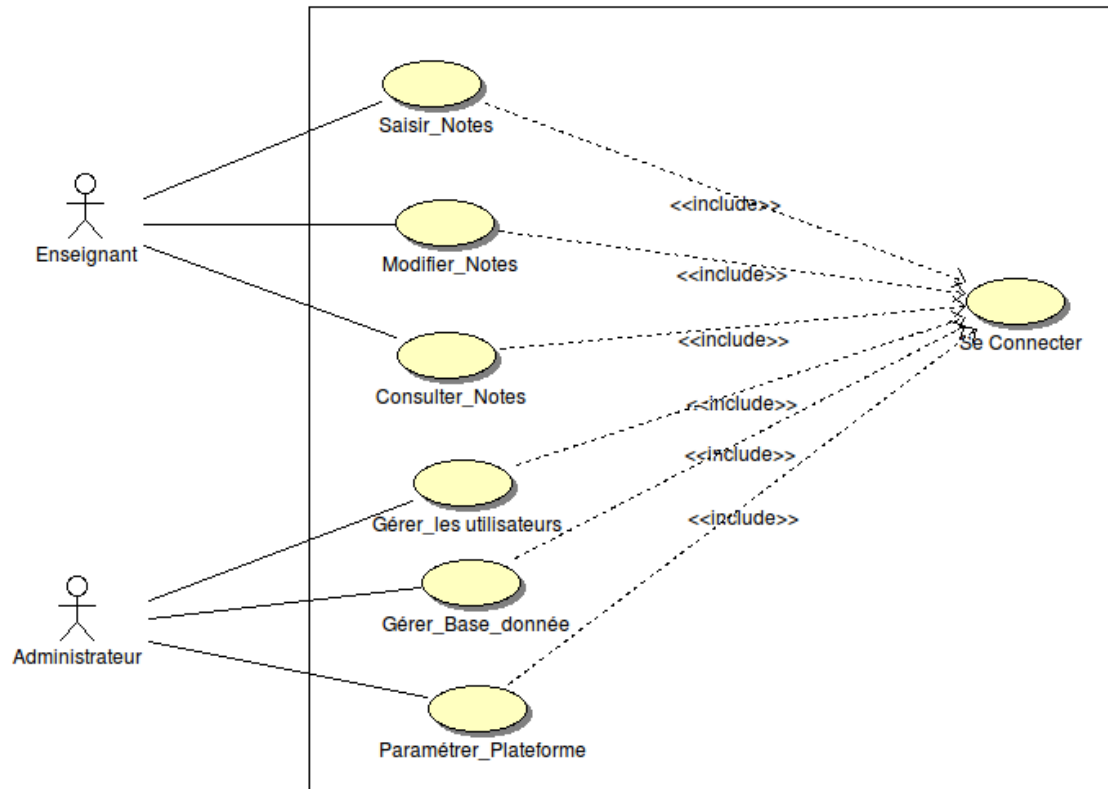
```
allow from localhost
```

```
</Directory>
```

```
</VirtualHost>
```

Ainsi au démarrage du serveur local vous arriverez directement sur l'accueil de l'application frontend.

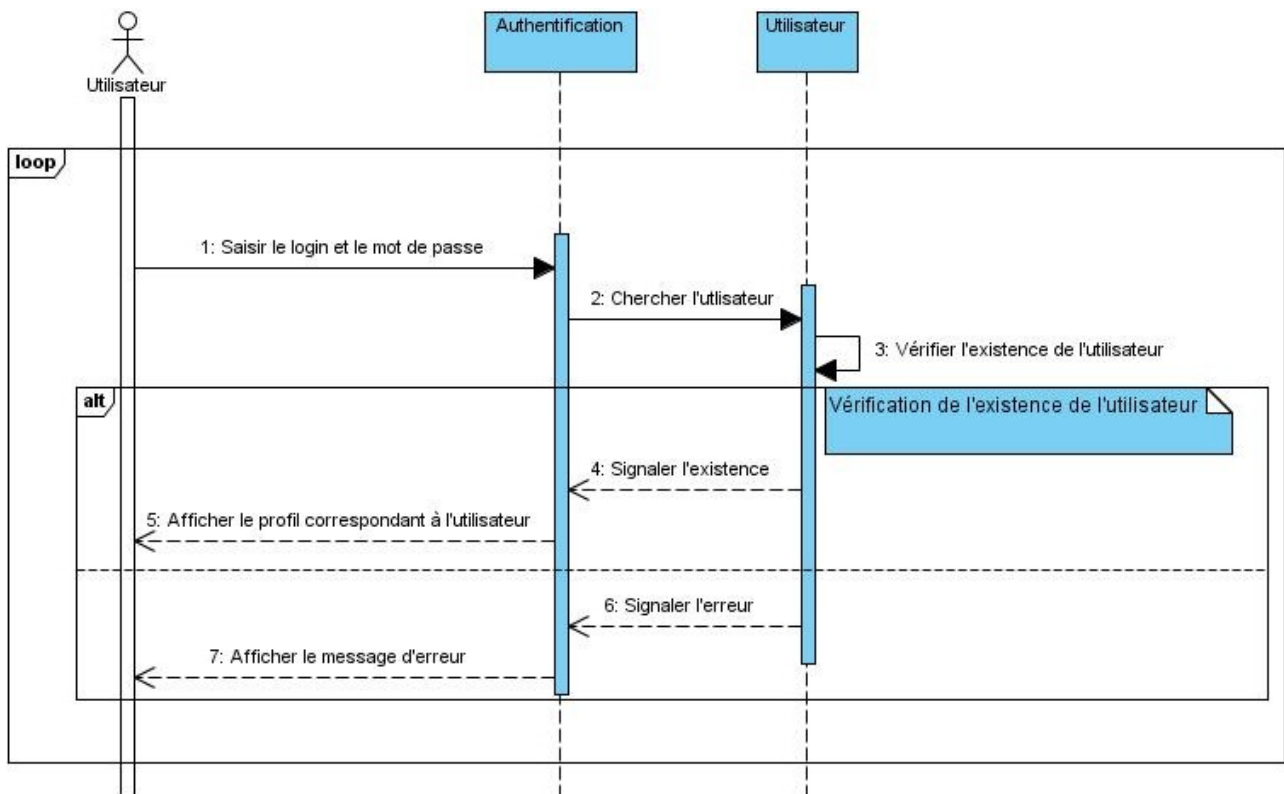
Diagramme de cas d'utilisation



L'enseignant, après s'être authentifié, pourra entrer les notes, modifier les notes déjà entrées, consulter les notes. L'administrateur, lui, après authentification, gère les utilisateurs, gère la base de données et peut paramétrer la plateforme.

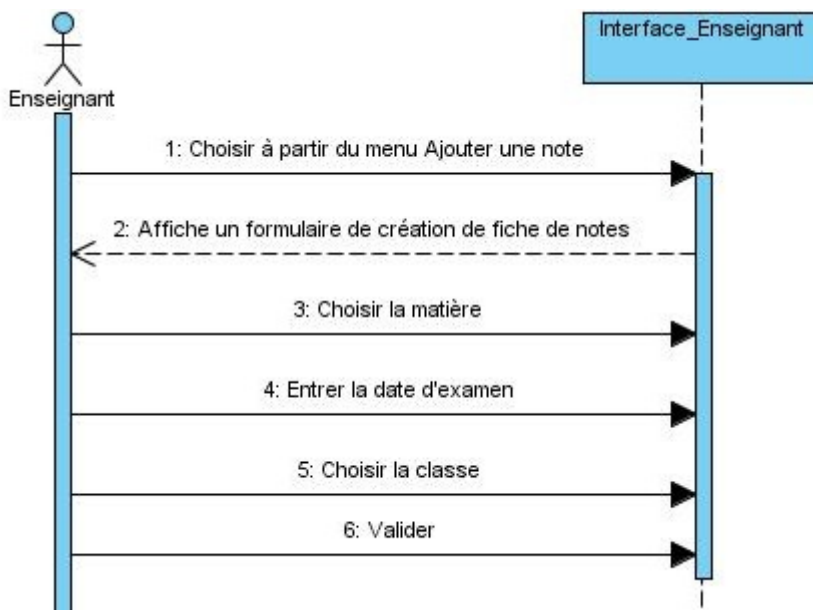
Diagrammes de séquence

a- Authentification



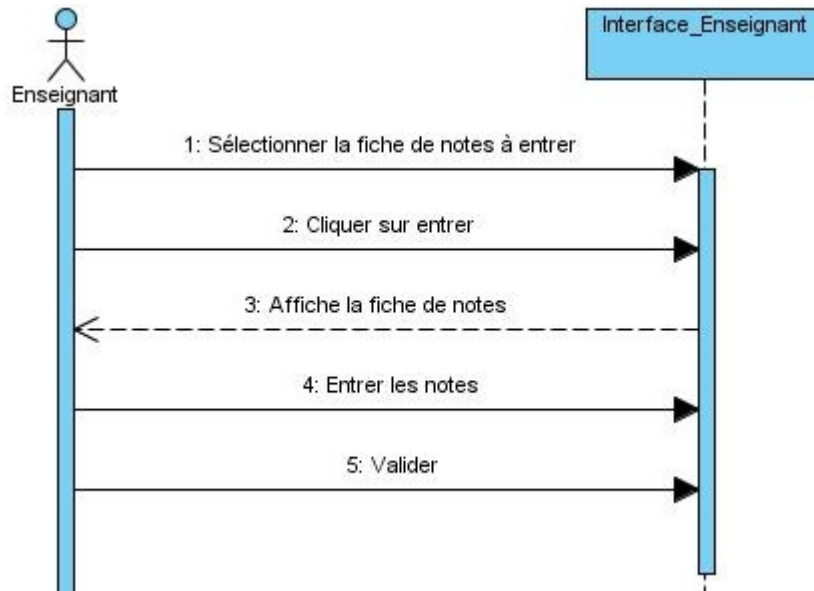
Après avoir lancé le site web, l'utilisateur (enseignant ou administrateur) saisit son login et mot de passe puis valide. Le système vérifie l'existence des informations entrées dans la base de données. Si elles existent alors le système affiche le profil correspondant à l'utilisateur qui s'est connecté, sinon il envoie un message d'erreur et demande à l'utilisateur de ressaisir le login et mot de passe.

b- Ajouter une fiche de notes



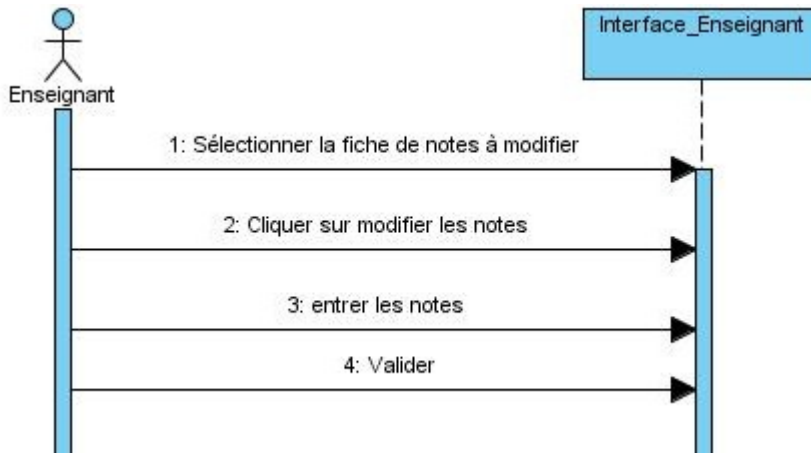
L'enseignant crée une fiche de notes qui lui permettra d'entrer les notes des étudiants, eues à un examen. Ensuite, il choisit à partir du menu << Ajouter une fiche de notes>>. Le système affiche un formulaire à remplir. Il choisit la matière, entre la date d'examen et choisit la classe puis il valide.

c- Ajouter une note



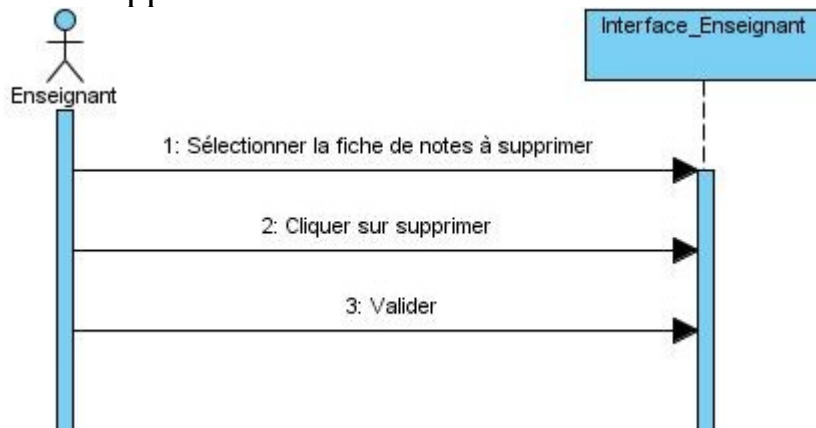
Pour entrer les notes, l'enseignant choisit la fiche de notes à entrer. Il clique ensuite sur le bouton <<entrer>>. Le système affiche ensuite la fiche de notes à remplir. L'enseignant entre, enfin, les notes et valide.

d- Modifier une note



Pour modifier certaines notes, l'enseignant sélectionne, en premier, la fiche de notes à modifier. Ensuite, il clique sur le bouton << modifier>>. Le système affiche la liste des notes. L'enseignant entre maintenant les vraies valeurs et valide.

e- Supprimer les notes



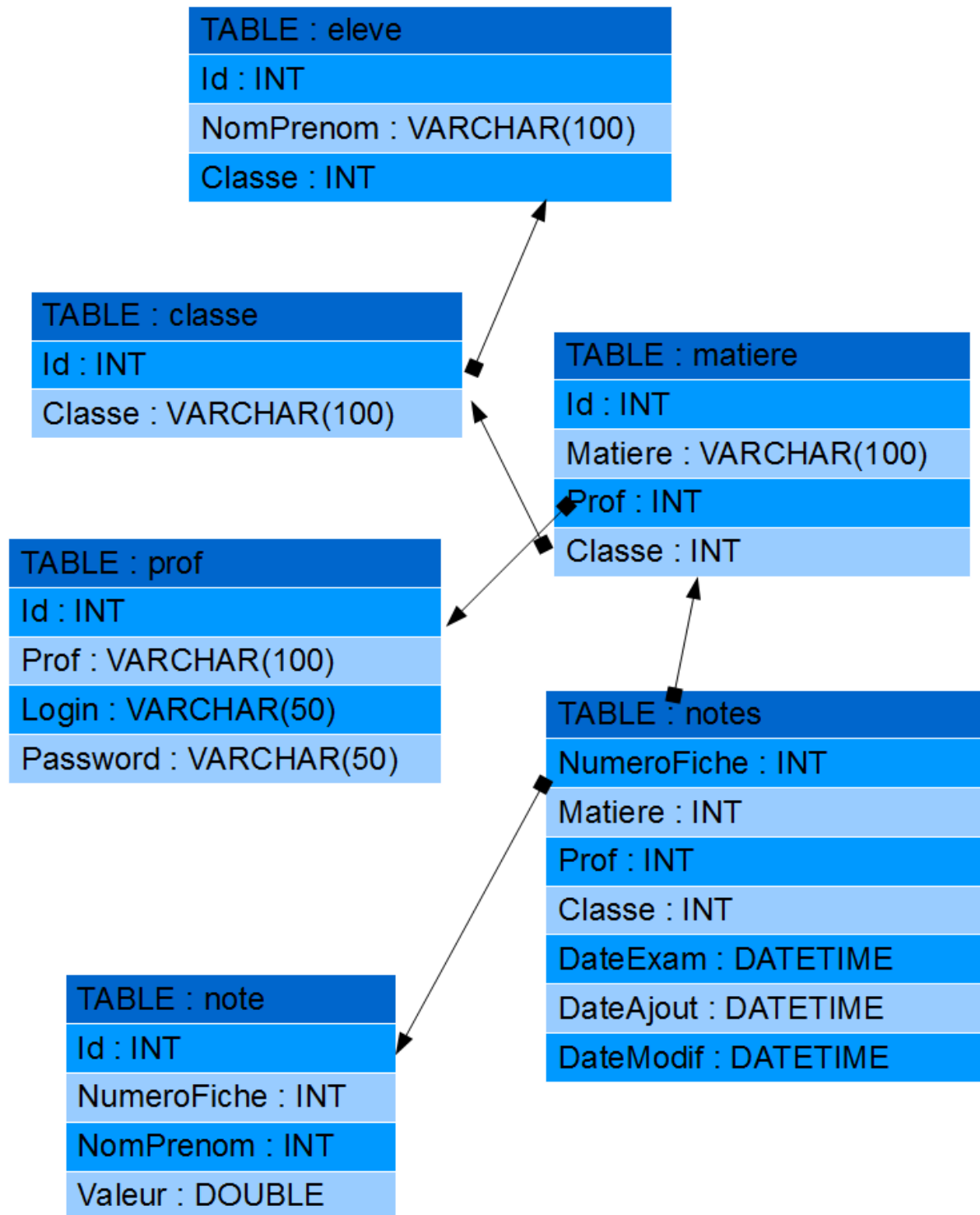
Pour supprimer des notes, l'enseignant sélectionne, en premier, la fiche de notes à modifier. Ensuite, il clique sur le bouton << supprimer>> et valide.

Base de données

La base de données va être composée de tables. Chaque entité utilisée sur le site aura sa table. Lors d'une requête, des jointures sont faites entre les tables pour obtenir toutes les informations.

Dans notre projet nous avons défini 6 tables : prof, élèves, matiere, classe, notes, note.

Schéma d'organisation de la base de données :



Organisation du modèle et des applications

1-Diagrammes des classes de la bibliothèque

Le projet a été pris d'utiliser une modélisation orienté objet pour ce site. Et en particulier utiliser un modèle MVC. La définition de l'ensemble du Framework, a été définie dans une bibliothèque. Une autre possibilité aurait été d'utiliser des Framework déjà existant.

On va décomposer le site internet en plusieurs applications. Chaque application correspondra au niveau d'accessibilité des utilisateurs. Chacune de ces applications sera décomposée en modules qui regroupent un ensemble de compétences.

Par exemple, l'application « Frontend » qui est la partie visible de tous du site, possède un module « Note » qui a pour compétences l'affichage des fiches de notes et l'affichage de l'ensemble des notes correspondant à une fiche.

Diagramme de classes de la bibliothèque :

Explication des classes de la bibliothèque applicative :

Application (abstraite) : c'est cette partie que l'on va exécuter, elle est caractérisée par son nom. Elle possède également la requête du client et la réponse à envoyer.

ApplicationComposant (abstraite) : cette classe représente tous les objets qui vont compléter l'application.

Classes filles de ApplicationComponent :

HTTPReponse : réponse à envoyer à l'utilisateur.

HTTPRequest : requête de l'utilisateur.

Contrôleur : c'est le contrôleur de l'application, c'est lui qui via une instance de Page va envoyer les données à la vue. Le contrôleur est propre à chaque module.

User : c'est l'utilisateur qui est connecté sur le site.

Page : représente la page web, le cadre de la page et la vue.

Routeur : il est caractérisé par un tableau de route. C'est lui qui en fonction de l'url va rediriger vers une route.

Route : c'est l'objet qui représente la page web à afficher. Il est caractérisé par une url. Il possède également un tableau de variable, que l'on récupère par l'url ou par des sessions.

Modèle : permettent l'interaction avec les données.

L'utilisation d'un tel modèle, facilite grandement le développement d'applications dans le site internet, et généralise le développement de celles-ci notamment dans la perspective de modification ultérieure du site.

2-diagrammes des classes des applications :

Dans cette partie, nous allons présenter la réalisation d'une application dans le site web en utilisant la bibliothèque défini dans la partie précédente.

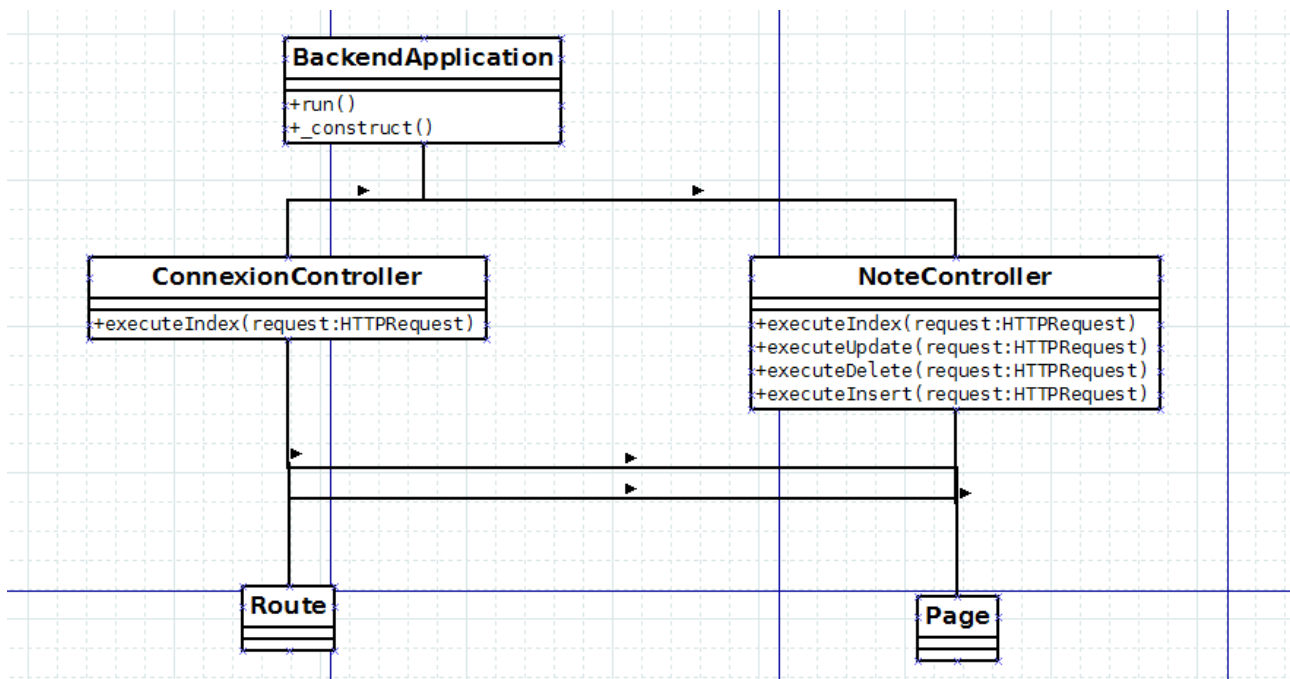
Les deux applications suivent exactement le même raisonnement. Nous nous concentrerons sur l'application « Backend » qui va contenir 2 modules, un pour la « Connexion » et un autre pour la gestion des « Notes ».

C'est l'application « BackendApplication » qui va implémenter une application. C'est lui qui va notamment assigner le contrôleur à utiliser.

On a donc deux contrôleurs « ConnexionController » et « NoteController ». Ces contrôleurs vont exécuter les actions et associer à ces actions des vues.

Afin de savoir qu'elle action doit être exécutée, c'est le routeur qui va être utilisé. En pratique ce routeur est un fichier xml qui contient toutes les routes de l'application.

A chaque exécution présente dans le contrôleur, correspond une vue, défini par l'objet Page.



3-Composants Web

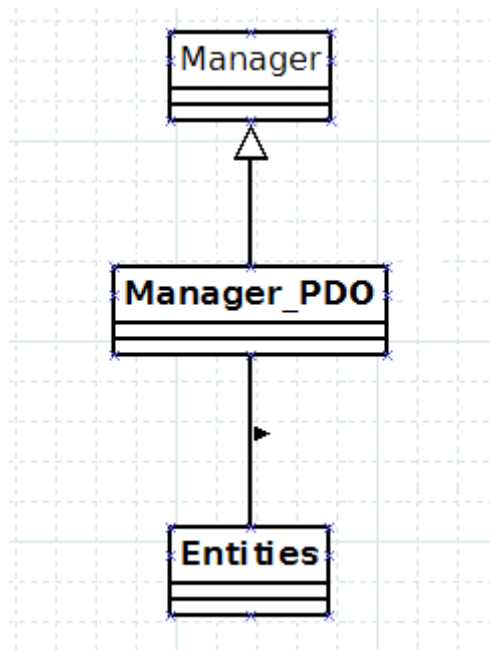
Dans le projet PHP, il existe aussi une partie « Web ». Qui contient principalement la charte graphique du site. Pour cette charte graphique, nous avons utilisé un package libre nommé « Envision ».

Concrètement chaque page du site est composé de deux parties. D'une part il y a une partie fixe, commune à toutes les pages de l'application, c'est le layout. La deuxième partie, est variable d'une action à une autre et c'est cette partie qui est plus particulièrement spécifiée dans les vues.

4- gestion des données

Tout au long du déroulement de l'application, des données sont traitées. Pour chaque table présente dans la base de données, correspond un objet (Classe, eleve, matiere, note, notes, prof). Afin d'utiliser ces formats et de récupérer des informations dans les tables correspondantes, c'est le « Manager » qui va intervenir. Et plus précisément ce « Manager » va être implémenté par un « Manager_PDO » qui lui ait préparé à communiquer avec la base de données.

Les informations relatives à la connexion à la base de données, sont à spécifier dans le PDOFactory.class.php de la bibliothèque.



Toutes ces données sont présentées dans le dossier du code, dans une arborescence particulière. Ce qui touche au traitement des données et le Framework sont dans le dossier « Library ». L'ensemble des composants Web sont dans le dossier « Web ». Enfin le site en lui-même est dans le dossier « Applications ». Dans ce dossier on crée à nouveau 2 arborescences une pour chaque application. Chacune de ces applications possède un dossier pour les fichiers de configuration (exemple : les routes), également un dossier pour le « layout », et à nouveau des arborescences pour les modules qui vont contenir le Contrôleur et un dossier qui contient les différentes vues.

<Schéma arborescence des dossiers>

Utilisation du site Web

1 – Utilisation de l'application Frontend

Comme mentionner précédemment le site web est composé de 2 applications. D'une part l'application frontend est accessible à tous. C'est l'application qui est lancé au début de l'exécution du site.

Concrètement il correspond à deux pages :

La page d'accueil qui affiche toutes les fiches de notes disponibles, dans un tableau.

Accueil Deconnex			
Professeur	Classe	Matière	Date de l'examen
Lenglart	GM5	Mesure	2013-12-30 22:27:59
Lenglart	GM5	Mesure	2013-12-30 22:27:59
ProfTest	GM5	Controle	2013-01-09 16:30:00
WRespondek	GM5	EquaDiff	2013-01-05 22:26:00
CLeGuyader	GM4	Bezier	2013-12-30 22:27:59
Lenglart	GM3	Mesure	2013-01-05 22:26:00
Lenglart	GM4	Markov	2013-12-30 22:27:59

Pour avoir les détails des fiches de notes, il suffit de cliquer sur la fiche désirée et on accède à la page avec l'ensemble des notes correspondantes.

Accueil	
Matière : 1 du 2013-12-30 22:27:59	
Date de saisie : 2013-01-10 11:16:20	
Comp	10
Alone	11

2-Utilisation de l'application Backend

La partie Backend correspond à la partie du site uniquement accessible aux professeurs possédant un mot de passe. Dans un premier temps, il y aura un module de connexion au site, puis à nouveau un module de gestion de notes.

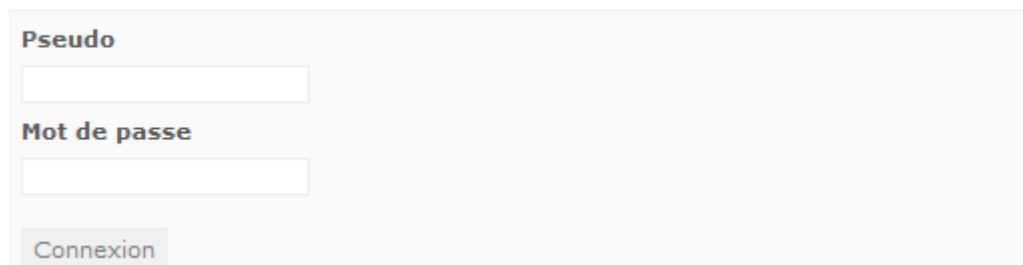
a) Module de connexion

Pour accéder à la page de connexion, l'utilisateur doit cliquer sur l'onglet « Admin » ou « ajouter une fiche » dans la barre des menus. (Pour se déconnecter il suffit d'utiliser l'onglet « deconnexion »)











On accède au formulaire de connexion. Chaque professeur s'est vu attribué un login et un mot de passe différent.

Connexion

Un formulaire de connexion sur un fond gris clair. Il contient deux champs de saisie : "Pseudo" et "Mot de passe", chacun avec un rectangle blanc pour le texte. En dessous, à gauche, se trouve un bouton "Connexion" avec un fond gris et du texte blanc.

Une fois les coordonnées saisies on accède à la partie administration du site, et en particulier à une page d'accueil, qui affiche la liste de toutes les fiches de notes existantes uniquement du professeur connecté. Pour chaque fiche 2 options sont disponibles, le « crayon » permet de modifier les notes de la fiche, et l'outil « croix rouge » permet de supprimer la fiche.

Il y a actuellement 7 notes. En voici la liste :

Numéro	Professeur	Matière	Date de l'examen	Date d'ajout	Dernière modification	Action
7	Lenglart	Mesure	2013-12-30 22:27:59	le 2013-01-10 11:16:20	-	 
6	Lenglart	Mesure	2013-12-30 22:27:59	le 2013-01-10 11:00:28	-	 
2	Lenglart	Mesure	2013-01-05 22:26:00	le 2013-01-09 18:20:54	-	 
1	Lenglart	Markov	2013-12-30 22:27:59	le 2013-01-09 18:20:44	-	 

Outil mise à jour :

Modifier une note : _____

Nom : **Comp** ▼

Note : _____

Ajouter

Comp 10

Alone 11

Lorsqu'un professeur souhaite saisir les notes d'un examen. Dans un premier temps il doit créer une nouvelle fiche de notes avec l'outil « Ajouter une notes » dans la barre des menus. Lorsque la fiche est créée automatiquement l'ensemble des élèves concernés ont attribué une note par défaut, 0. Le professeur doit ensuite utiliser l'outil de mise à jour, pour modifier les notes. Un professeur ne peut créer une fiche que pour l'une de ses matières.

Création fiche de notes :

Création de la fiche : _____

Professeur : 1

Matiere : **Mesure** ▼

Date Exam (format : AAAA-MM-JJ HH:MM:SS): _____

Classe : **GM5** ▼

Ajouter

Conclusion

Au cours de ce projet nous avons eu l'occasion d'approfondir nos connaissances en modélisation objet et en tenant de réaliser un modèle MVC. Nous avons notamment eu l'occasion d'utiliser ces connaissances pour la réalisation d'un site web en php.

L'utilisation du framework doit rendre par l'avenir très simple l'utilisation de nouveaux types de données, et l'intégration de nouvelles compétences au site web.

Des extensions possibles peuvent être à prévoir pour la saisie de notes, notamment l'utilisation d'un formulaire qui récupère un tableur Excel ou csv et insère les notes automatiquement dans la base.