**REPUBLIQUE DU SENEGAL**



**UNIVERSITE CHEIKH ANTA DIOP DE DAKAR**



**ECOLE SUPERIEURE POLYTECHNIQUE**

***DEPARTEMENT GENIE INFORMATIQUE***

**MEMOIRE DE FIN DE CYCLE**

**Pour l’obtention du :**

DIPLOME UNIVERSITAIRE DE TECHNOLOGIE (DUT)

**SUJET :**

***REFONTE ET EXTENSION DU SITE WEB DE L’ESP : CREATION D’ESPACES ETUDIANT, ENSEIGNANT ET SECRETAIRE ET ADMINISTRATEUR***

**Lieu de stage** : CRENT

**Période de stage : 04/06/2018 – 20/07/2018**

|  |  |  |
| --- | --- | --- |
| **Présenté et soutenu par** | **Professeur encadrant** | **Maitre de stage** |

Dr Mamadou CAMARA

Jean Philippe FALL

Aliou GUEYE

Mme. NGOM Bineta DIALLO

**REPUBLIQUE DU SENEGAL**



**UNIVERSITE CHEIKH ANTA DIOP DE DAKAR**



**3**

**ECOLE SUPERIEURE POLYTECHNIQUE**

***DEPARTEMENT GENIE INFORMATIQUE***

**MEMOIRE DE FIN DE CYCLE**

**Pour l’obtention du :**

DIPLOME UNIVERSITAIRE DE TECHNOLOGIE (DUT)

**SUJET :**

***REFONTE ET EXTENSION DU SITE WEB DE L’ESP : CREATION D’ESPACES ETUDIANT, ENSEIGNANT ET SECRETAIRE***

**Lieu de stage** : CRENT

**Période de stage : 04/06/2018 – 20/07/2018**

|  |  |  |
| --- | --- | --- |
| **Présenté et soutenu par** | **Professeur encadrant** | **Maitre de stage** |
| Jean Philippe FALL  Aliou GUEYE | Dr. Mamadou Samba CAMARA | Mme. NGOM Bineta DIALLO |

***DEDICACES***

Jean Philippe FALL

Je dédie ce présent mémoire :

A ma très chère mère **Emilie Marie Noelle MBAYE**,

Tu t’es toujours battue sans relâche notre bonheur et notre bien-être. Que le Seigneur puisse te conférer une longue vie et te permette de vivre et célébrer ma réussite.

A ma tendre grand-mère **Anna CISS**,

Depuis ma tendre enfance, tu m’as inculqué des valeurs morales très importantes qui m’ont permis de ne jamais laisser les bras, de me battre jusqu’au bout afin de réaliser mes rêves.

A mon cher frère **Gilbert MBAYE**,

Mon confident, mon partenaire de toujours.

**A Mouhamadou Ahmadou Ciré « Dédé » DIA**,

Mon mentor de toujours, tu as su réveiller la passion informatique qui dormait en moi.

A mes camarades de chambres **Abdou Khadre Dieylani DRAME**, **Mamadou Mbacké FALL** et **Khadim « Lord » GUEYE**,

Aliou GUEYE

Au nom d’**Allah le très Miséricordieux**

Je dédie ce travail à :

A mon père **Dame GUEYE** et à ma mère **Ndeye Anta DIOP**

Vous avez su être pour moi des parents de référence qui m'ont toujours aidé dans ce que j'entreprends et qui m'ont inculqué une éducation de qualité.

A mes très chers frères et sœurs que j'estime beaucoup.

Mes camarades de classe et mes camarades de stage.

A tous les membres du **Daara MAJMAHUN NORAYNI,**

***REMERCIEMENTS***

Nous remercions tout d’abord Le Tout-Puissant qui nous a permis de mener à bien le sujet soumis à notre étude. Nos remerciements s’adressent aussi particulièrement à :

* Nos parents respectifs pour l’accompagnement et la motivation que vous nous insufflez chaque jour,
* Notre Me de stage **Mme. Bineta DIALLO** pour sa disponibilité, son ouverture et son sens de la compréhension qui nous mettent dans une bonne ambiance de travail,
* Notre encadreur, **M. Mamadou CAMARA**, pour sa disponibilité, sons sens critique et ses recommandations dont il ne s’est lassé de nous procurer,
* L’**ensemble du personnel du CRENT** pour leur jovialité et leur gentillesse,
* Nos **promotionnaires**,
* Nos filleuls, parrains, marraines et mames du campus social,
* L’ensemble des personnes ayant contribué de près ou de loin à la réalisation de ce présent document.

***Avant-propos***

L’Université Cheikh Anta Diop de Dakar (UCAD), prestigieuse université de la capitale sénégalaise, comporte plusieurs écoles et instituts sous sa tutelle parmi lesquels l’Ecole Supérieure Polytechnique de Dakar (ESP).

L’Ecole Supérieure Polytechnique de Dakar, établissement public à vocation interafricaine, a pour mission de former des techniciens supérieurs (bac +2) et des ingénieurs (bac +5) tant sur le plan théorique que pratique. Les étudiants en fin de formation doivent effectuer un stage obligatoire de durée moyenne pour l’obtention du Diplôme Universitaire de Technologie (DUT) suivi de la présentation d’un rapport de stage.

Le stage permet en effet aux étudiants :

- De découvrir les réalités de l’univers du travail notamment en entreprise,

- D’évaluer leurs connaissances acquises au cours des deux années de formation,

- De renforcer leurs capacités dans la conception et la réalisation de solutions informatiques relatives aux besoins de l’entreprise.

Le mémoire, qui est un support de communication, permet aux étudiants de mettre en relation la théorie et la pratique et de présenter le travail effectué en stage à leur établissement.

Selon ce but, nous avons effectué un stage d’un mois et quinze jours au niveau du Centre des Ressources pour l’Environnement Numérique de Travail (CRENT) pour la refonte du site web de l’Ecole Supérieure Polytechnique portant plus précisément sur la création d’un espace étudiant, d’un espace enseignant et d’un espace secrétaire.

Dans ce rapport, il sera question de parler du déroulement du stage, de la conception et de la réalisation de notre solution, et des difficultés rencontrées.

***Table des matières***

**INTRODUCTION**

**Chapitre 1 : PRESENTATION GENERALE**

1. Présentation de l’ESP

I.1. Historique

I.2. Missions

I.3. Les départements

1. Présentation du CRENT

II.1. Rôles du CRENT

II.2. Missions du CRENT

II.3. Organigramme du CRENT

1. Présentation du sujet

III.1. Contexte

III.2. Problématique

III.3. Objectifs

1. Conclusion

**Chapitre 2 : ANALYSE DES BESOINS ET MODELISATION DE LA SOLUTION**

1. Processus de développement et langage UML

I.1. Processus de développement

I.2. Présentation du langage UML

1. Analyse et spécification des besoins

II.1. Spécifications des besoins et des acteurs

II.2. Spécification des fonctionnalités

1. Diagrammes de cas d’utilisation
2. Fiches textuelles et diagrammes de séquence
3. Diagramme de classes
4. Conclusion

**Chapitre 3 : CONCEPTION ET MISE EN ŒUVRE DE L’APPLICATION**

1. Conception

I.1. Architecture 3 tiers

I.2. Langages et technologies utilisés

I.3. Déploiement de la base de données

1. Implémentation de l’application

II.1. Environnement de développement

II.2. Réalisation de la plateforme

**CONCLUSION**

***Table des figures***

[Figure 1 Organigramme hiérarchique du CRENT 14](file:///C:\Users\jpfal\Desktop\New%20folder\Mémoire%20AG%20JPF.docx#_Toc521286674)

[Figure 2 Versions d'UML 17](#_Toc521286675)

[Figure 3 Hiérarchisation des diagrammes UML 19](#_Toc521286676)

[Figure 4 Diagramme de cas d’utilisation de l’étudiant 21](#_Toc521286677)

[Figure 5 Diagramme de cas d’utilisation de l’enseignant 21](#_Toc521286678)

[Figure 6 Diagramme de cas d’utilisation de la secrétaire 22](#_Toc521286679)

[Figure 7 Diagramme de cas d’utilisation de l’administrateur 23](#_Toc521286680)

[Figure 8 Diagramme de séquence du cas « S’authentifier » 24](#_Toc521286681)

[Figure 9 Diagramme de séquence du cas « Consulter emploi du temps » 26](#_Toc521286682)

[Figure 10 Diagramme de séquence du cas « Consulter notes » 27](#_Toc521286683)

[Figure 11 Diagramme de séquence du cas « Enregistrer notes » 28](#_Toc521286684)

[Figure 12 Diagramme de séquence du cas « Ajouter enseignant » 29](#_Toc521286685)

[Figure 13 Diagramme de classes 31](#_Toc521286686)

[Figure 14 Illustration des 3 couches de l’architecture 3-tiers 32](#_Toc521286687)

[Figure 15 Représentation d’une application web avec l’architecture 3 tiers 34](#_Toc521286688)

[Figure 16 Base de données de l’application 37](#_Toc521286689)

[Figure 17 Page d’authentification du site 41](#_Toc521286690)

[Figure 18 Page d’authentification du site 42](#_Toc521286691)

[Figure 19 Page de consultation des notes 43](#_Toc521286692)

[Figure 20 Page de consultation de l’emploi du temps 43](#_Toc521286693)

[Figure 21 Page de consultation des absences 44](#_Toc521286694)

[Figure 22 Page de consultation de l’information 44](#_Toc521286695)

[Figure 23 Page de saisie d’absence (Par l’enseignant) 45](#_Toc521286696)

[Figure 24 Page de mise à jour des notes des étudiants 46](#_Toc521286697)

[Figure 25 Page de mise à jour des classes 47](#_Toc521286698)

[Figure 26 Page de publication de l’information 48](#_Toc521286699)

[Figure 27 Page d’ajout d’une secrétaire 49](#_Toc521286700)

[Figure 28 Page d’ajout d’une secrétaire 50](#_Toc521286701)

[Figure 29 Page de listage des secrétaires 50](#_Toc521286702)

***INTRODUCTION***

L’Ecole Supérieure Polytechnique de Dakar, selon les statistiques en 2015, comptait un total de 4159 étudiants régulièrement inscrits. Ce chiffre s’en va, sans doute, augmenter d’années en années. Cependant, dans la gestion des étudiants, ces derniers ont toujours été des acteurs passifs, n’interagissant pas avec le système de gestion des étudiants de manière active. Ils se retrouvent dès lors écartés d’un système de gestion où leur participation pourrait encore plus prendre de l’envergure. D’où la nécessité de la création d’un espace étudiant qui leur permettrait d’interagir avec ce système.

En effet, la création d’un espace étudiant au sein du site web de l’ESP, qui est un site à accès public, pourrait optimiser l’interaction entre les différentes entités concernées par la gestion des étudiants de l’Ecole Supérieure Polytechnique de Dakar mais aussi permettre aux étudiants de mieux se retrouver dans la vie pédagogique de l’établissement. Le choix du site de l’ESP est favorisé par sa longue extension à tous les départements de l’établissement.

Ainsi il nous a été assigné la tache de réaliser cette solution informatique lors de notre stage d’une durée d’un mois et demi au Centre des Ressources pour l’Environnement Numérique de Travail (CRENT).

Nous allons articuler notre document autour de 3 chapitres :

* Chapitre 1 : Présentation générale
* Chapitre 2 : Analyses des besoins et modélisation de la solution
* Chapitre 3 : Conception et mise en œuvre de l’application

**CHAPITRE**

**1**

**PRESENTATION GENERALE**

**I. PRESENTATION DE L’ESP**

**I.1. Historique**

L’Ecole Supérieure Polytechnique de Dakar (ESP) est un établissement à la fois public et privé à vocation régionale doté de la personnalité juridique et de l’autonomie financière.

L’ESP fait partie intégrante de l’Université Cheikh Anta Diop de Dakar (UCAD) et ses locaux sont implantés aussi bien à Dakar que dans la ville de Thiès. Elle a été créée par la loi n°94-78 du 24 Novembre 1994.

L’ESP regroupait en son sein :

* La division industrielle de l’Ecole Nationale Supérieure Universitaire de Technologie (ENSUT).
* L’Ecole Polytechnique de Thiès (EPT).
* La section technique industrielle de l’Ecole Normale d’Enseignement Technique et Professionnel (ENSEPT).

Cependant suite à la revitalisation de l’ENSEPT, la création de l’Université de Thiès et le rattachement de l’Institut Supérieure de Gestion de l’ESP, l’Ecole Supérieure Polytechnique est présentement composée uniquement de la division industrielle et de la division tertiaire de l’ex ENSUT. Elle est composée de deux centres :

* Le centre de Dakar (ex ENSUT/ENSEPT)
* Le centre de Thiès (ex EPT)

**I.2. Missions**

L’ESP a pour mission :

* De former aussi bien sur le plan théorique que pratique des techniciens supérieures titulaires d’un Diplôme Universitaire de Technologie (DUT, DST), des ingénieurs d’exécution titulaires d’un Diplôme d’Ingénieur Technologue (DIT), des ingénieurs de conception de titulaire d’un Diplôme d’Ingénieur de Conception (DIC) mais aussi des managers en gestion d’entreprise.
* Elle a aussi pour mission de dispenser un enseignement supérieur et des activités de recherche en vue de préparer directement aux fonctions d’encadrement dans la production, la recherche appliquée et les services.
* D’organiser des enseignements et des activités de recherche visant au perfectionnement permanent, à l’adaptation et à la participation, à l’évolution scientifique et technologique de même que celle managériale et économique.
* De procéder à des expertises dans le cadre de la formation à l’intention des entreprises publiques et privées.

**I.3. Les départements**

L’ESP compte six (6) départements :

* Le département Génie Chimique et Biologie Appliquée ;
* Le département Génie Civil ;
* Le département Génie Electrique ;
* Le département Génie Informatique ;
* Le département Génie Mécanique ;
* Le département Gestion.

**II. PRESENTATION DU CRENT**

**II.1. Rôle du CRENT**

Le Centre de Ressources pour l’Environnement Numérique de Travail (CRENT) assure la disponibilité de ressources informatiques tant du point de vue réseau qu’applicatif. Il met à la disposition du personnel, des étudiants et des usagers de l’établissement ses équipements (serveurs, visioconférence, vidéoprojecteurs, sonorisation, etc.) et ses locaux (salles). Il participe aussi à la communication de l’école à travers le site web, la messagerie électronique, les listes de diffusion, les réseaux sociaux et la messagerie mobile (Web SMS).

**II.2. Missions du CRENT**

#### Plus généralement, le CRENT assure la maîtrise d’œuvre du Système d’information de l’ESP en garantissant :

#### La mise à disposition d’applications de gestion ;

#### La gestion du réseau informatique, en assurant un accès sécurisé et fluide aux différents types d’utilisateurs ; La maintenance des postes de travail et des serveurs ;

#### La mise en place des plates-formes pédagogiques et documentaires (plateforme d’enseignement à distance, bibliothèque virtuelle numérique, plateforme de partage, …) ;

#### La gestion de la messagerie ;

#### La gestion et la mise à jour du site internet.

**II.3. Organigramme du CRENT**

#### Le Centre des Ressources pour l’Environnement Numérique de Travail est constitué des cellules suivantes :

#### Coordonnateur : Coordination Administration

#### Cellule Réseau et Maintenance : Réseau Maintenance Assistance

#### Cellule Application : Application Site web Messagerie Assistance

#### Cellule multimédia et communication numérique : Infographie Images et vidéos Réseaux sociaux Matériels

#### Cellule d’ingénierie pédagogique : Ingénierie Pédagogique Plateformes de formation et documentaire FOAD

#### L’organigramme hiérarchique du CRENT est illustré par la figure suivante

Figure 1 Organigramme hiérarchique du CRENT



Coordonnateur



Cellule Exploitation

et Support

Utilisateurs



Responsable

Exploitation et

Support

utilisateurs



Cellule Réseau et

Maintenance



Responsable

Réseau



Responsable

Maintenance



Cellule Application



Responsable

Administration

Applications



Responsable

Support

Applications



Cellule Multimédia

et Communication

Numérique



Responsable

Multimédia

**III. PRESENTATION DU SUJET**

**III.1. Contexte**

L’Ecole Supérieure Polytechnique de Dakar est un établissement dispensant des formations publiques et privées. Elle regroupe en son sein plus de 4000 étudiants.

Dans sa politique d’organisation, la gestion des étudiants (gestion des absences, gestion de notes, gestion de l’informations, …) revient exclusivement aux départements qui la composent. Chaque département possède son propre outil de gestion de ses étudiants tant de la formation privée que publique.

Il s’agira ici de reprendre les mêmes systèmes de gestion, qui sont presque tous similaires, sous forme d’une plateforme qui les intègrent le mieux tout en tenant compte des outils préexistants.

**III.2. Problématique**

Aujourd’hui, malgré une bonne gestion des étudiants des différents départements que compte l’ESP, les étudiants ont beaucoup de mal à accéder aux informations relatives à leur vie pédagogique. Ils peuvent passer un semestre sans connaître l’intégralité de leurs notes dans les différentes unités d’enseignement où ils sont évalués. De plus, ils ont plus ou moins du mal à se situer sur leurs absences et leurs emplois du temps et autres informations relatives aux activités pédagogiques de l’école. Dès lors se pose un problème de visibilité de l’information car il n’existe aucun outil numérique qui puisse permettre aux étudiants d’avoir une vue sur ces informations précédemment citées (notes, absences, communiqués, notes de services, etc.). Alors, étant donné les différents acteurs qui interviennent dans la gestion des étudiants, la création d’un outil numérique intégrant activement les étudiants s’annonce complexe. Ainsi nous nous posons la question suivante : comment solutionner le problème ici posé ?

**III.3. Objectif**

Partant des problèmes énoncés ci-dessus, nous nous sommes donnés comme objectif principal de proposer une solution réelle qui intègre les différents acteurs et répond aux besoins de ceux-ci de façon optimale par l’extension du site web de l’Ecole Supérieure Polytechnique, la création d’espace de connexion des différents acteurs concernés.

**IV. Conclusion**

Dans ce précédent chapitre, nous avons fait une présentation de l’ESP, du CRENT et du sujet de notre stage, nous allons ensuite passer à l’analyse des besoins et la modélisation de la solution dans le chapitre suivant.

**CHAPITRE**

**2**

**ANALYSE DES BESOINS ET MODELISATION DE LA SOLUTION**

Ce présent chapitre consiste à faire une analyse succincte des besoins et une approche modélisée de la solution. Ici, nous sommes appelés à faire l’étude des acteurs, de leurs besoins et de proposer un modèle de solution à leurs attentes.

1. **PROCESSUS DE DEVELOPPEMENT ET LANGAGE UML**

**I.1. Processus de développement**

Pour la réalisation de notre application, nous avons plutôt opté pour la **méthode du cycle en V**. Ce modèle en cycle de développement permet de limiter les retours aux étapes précédentes, ce qui constitue un énorme avantage dans les circonstances d’un stage de moyenne durée où la bonne gestion du temps compte. La méthode du cycle en V implique aussi, avant chaque phase du projet, des réunions de lancement (kick off). Cette méthode permet alors d’anticiper et de préparer les résultats attendus pour les étapes futures.

La revue du projet est ainsi facilitée.

**I.2. Présentation du langage UML**

Le **Langage de Modélisation Unifié**, de l'anglais Unified Modeling Language (**UML**), est un [langage](https://fr.wikipedia.org/wiki/Langage) de modélisation à base de [pictogrammes](https://fr.wikipedia.org/wiki/Pictogramme) (représentation graphique schématique) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en [développement logiciel](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_logiciel) et en [conception orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet).

L'UML est le résultat de la fusion de précédents langages de modélisation objet : [Booch](https://fr.wikipedia.org/wiki/M%C3%A9thode_Booch), [OMT](https://fr.wikipedia.org/wiki/Object_Modeling_Technique), [OOSE](https://fr.wikipedia.org/wiki/OOSE). Principalement issu des travaux de [Grady Booch](https://fr.wikipedia.org/wiki/Grady_Booch), [James Rumbaugh](https://fr.wikipedia.org/wiki/James_Rumbaugh) et [Ivar Jacobson](https://fr.wikipedia.org/wiki/Ivar_Jacobson), UML est un standard adopté par l'[Object Management Group](https://fr.wikipedia.org/wiki/Object_Management_Group) (OMG) depuis 1997.

UML est un moyen d'exprimer des modèles objet en faisant abstraction de leur implémentation, c'est-à-dire que le modèle fourni par UML est valable pour n'importe quel langage de programmation.

UML est langage qui évolutif comme illustré par l’image suivante selon le site de l’OMG.



Figure 2 Versions d'UML

* **Les diagrammes en UML**

Le méta modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet ainsi que les liens qui les relie. Les versions d’UML antérieures à UML 2.0 proposaient neuf (9) diagrammes. UML 2.0 en a rajouté quatre (4). Et depuis UML 2.3, il en existe désormais quatorze (14). Ces quatorze types de diagrammes distinguent les vues pour représenter des concepts particuliers du système d'information. Ils se classent en trois ensembles :

* **Les diagrammes structurels ou statiques :**
* **Diagramme de classes :** qui montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc. Il représente les classes intervenant dans le système ;
* **Diagramme d’objets :** qui montre les instances des éléments structurels et leurs liens à l’exécution ;
* **Diagramme de paquetages :** qui présente l’organisation logique du modèle et les relations entre packages ;
* **Diagramme de structure composite :** montre l’organisation interne d’un élément statique complexe ;
* **Diagramme de composants :** qui montre des structures complexes, avec leurs interfaces fournies et requises ;
* **Diagramme de déploiement :** qui montre le déploiement physique des « artefacts » sur les ressources matérielles ;
* **Diagramme de profil :** depuis UML 2.2, permet de spécialiser, de personnaliser pour un domaine particulier un méta-modèle de référence d’UML ;
* **Les diagrammes comportementaux :**
* **Diagramme de cas d’utilisation :** qui montre les interactions fonctionnelles entre les acteurs et le système à l’étude ;
* **Diagramme états-transitions :** qui montre les différents états et transitions possibles des objets d’une classe ;
* **Diagramme d’activité :** qui présente l’enchaînement des actions et décisions au sein d’une activité ;
* **Diagrammes d’interaction ou dynamiques :**
* **Diagramme de séquence :** qui montre la séquence verticale des messages passés entre objets au sein d’une interaction ;
* **Diagramme de communication :** qui montre la communication entre objets dans le plan au sein d’une interaction ;
* **Diagramme global d’interaction :** qui fusionne les diagrammes d’activité et de séquences pour combiner des fragments d’interactions avec des décisions et des flots ;
* **Diagramme de temps :** qui fusionne les diagrammes d’états et de séquence pour montrer l’évolution de l’état d’un objet au cours du temps ;

La figure suivante représente l’ensemble des quatorze diagrammes que comporte le langage UML ainsi que leur catégorisation.

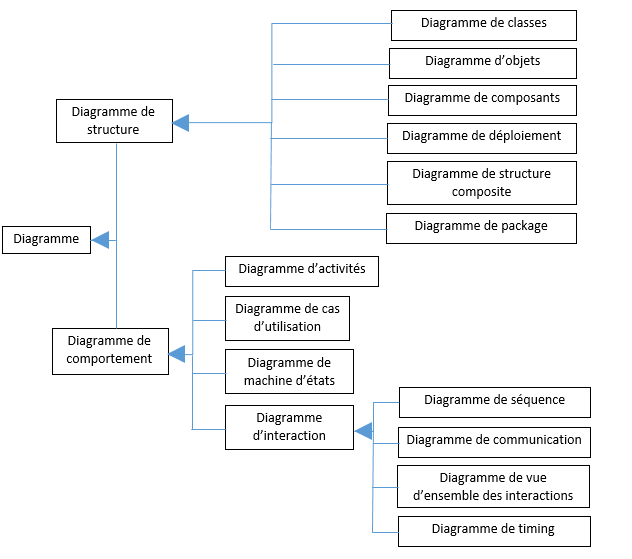


Figure 3 Hiérarchisation des diagrammes UML

1. **ANALYSE ET SPECIFICATIONS DES BESOINS**

**II.1. Spécifications des besoins et des acteurs**

Le système que nous allons mettre en place nécessite une certaine sécurité, par conséquent tout acteur devra s’authentifier pour avoir un accès sécurisé à son compte personnel. Le système va se composer essentiellement de quatre (4) acteurs à savoir l’administrateur, l’étudiant, l’enseignant et la secrétaire.

L’étudiant, dans ce système, peut consulter l’ensemble de ses notes qui sont disponibles sur la plateforme, mais aussi consulter ses absences de manière à s’informer sur le jour et la matière concernée, voir les informations à caractère pédagogique et enfin visualiser son emploi du temps.

L’enseignant a la possibilité d’enregistrer les absences liées à son cours et aussi d’enregistrer les notes d’évaluations pour chaque étudiant.

La secrétaire peut mettre à jour l’ensemble des étudiants, ou l’ensemble des enseignants, ajouter une classe et publier de l’information utile aux étudiants.

L’administrateur est capable d’ajouter, mettre à jour ou supprimer un département. Il peut aussi mettre à jour, ajouter ou supprimer une secrétaire.

**II.2. Spécification des fonctionnalités**

La conception d’une application d’une telle envergure nécessite une analyse approfondie pour une spécification sans faille des fonctionnalités des différents acteurs. A cet effet, nous mènerons notre spécification des fonctionnalités au moyen des diagrammes de cas d’utilisations, des diagrammes de séquences et de fiches textuelles pour chaque acteur de la plateforme. L’analyse, quant à elle, sera illustrée à l’aide d’un diagramme de classes.

1. **Diagrammes de cas d’utilisations**

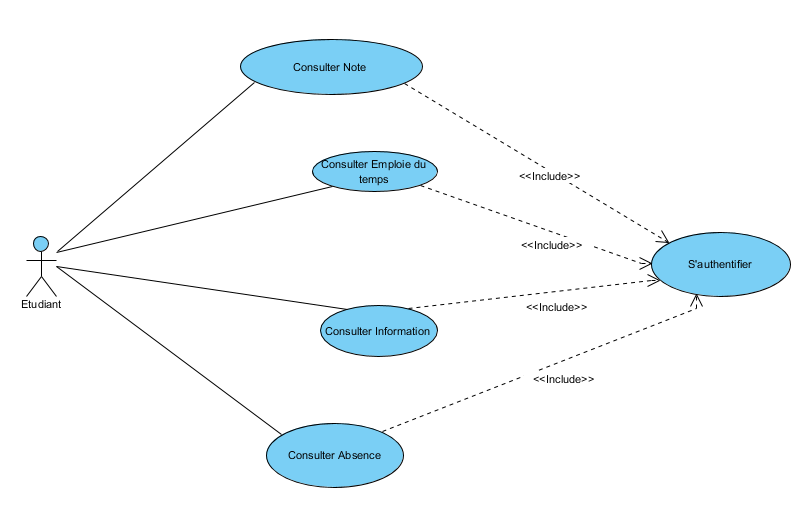


Figure 4 Diagramme de cas d’utilisation de l’étudiant

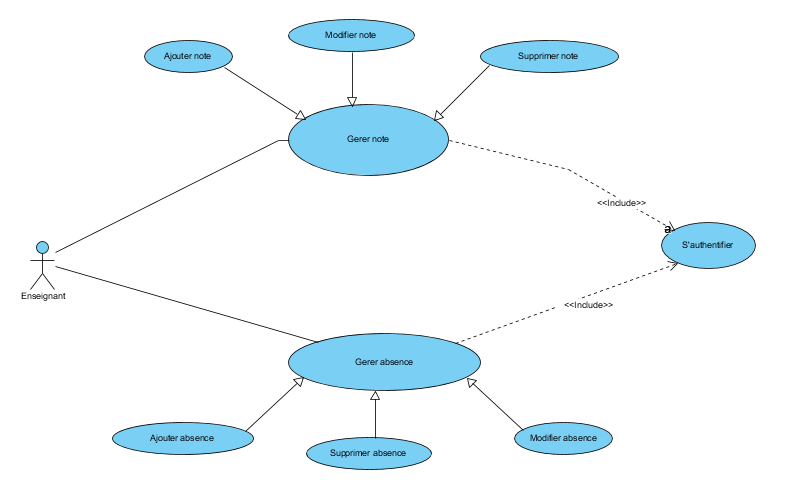


Figure 5 Diagramme de cas d’utilisation de l’enseignant

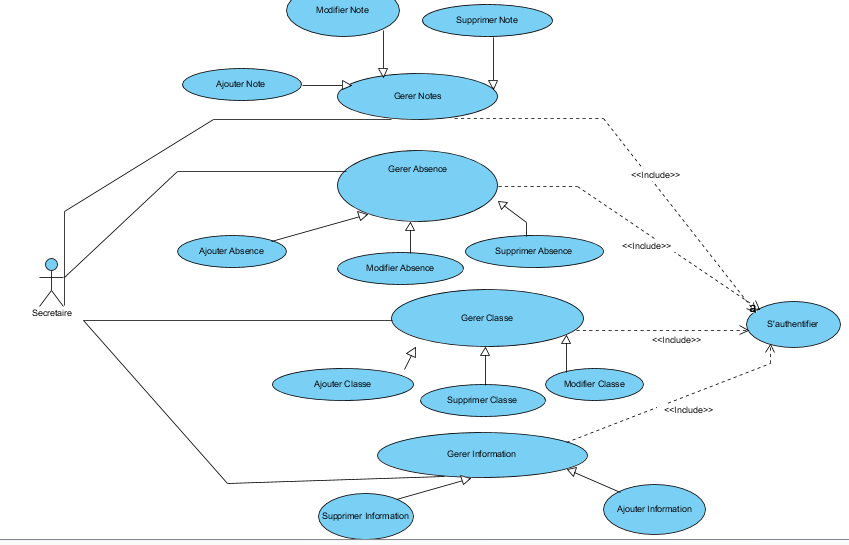


Figure 6 Diagramme de cas d’utilisation de la secrétaire

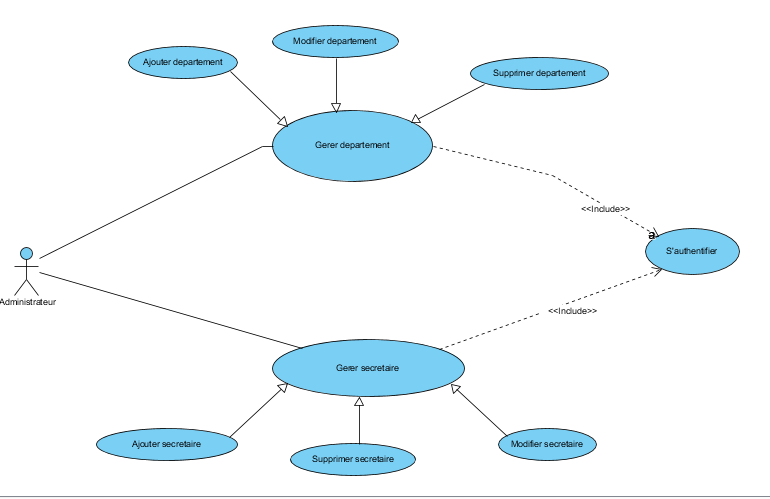


Figure 7 Diagramme de cas d’utilisation de l’administrateur

1. **Fiches textuelles et Diagrammes de séquences**

* **Cas « S’authentifier » :**

|  |  |
| --- | --- |
| Cas | S’authentifier |
| Objectif | Permettre à un utilisateur d’accéder à son compte sur la plateforme |
| Acteur(s) | Tout type d’utilisateur (Administrateur, Enseignant, Etudiant, Secrétaire) |
| Précondition | Accéder à la page d’authentification de l’application |
| Scénario nominal | 1. Entrer son login et son mot de passe 2. Valider le login et le mot de passe 3. Vérification du système 4. Le système redirige vers la page d’accueil du compte de l’utilisateur |
| Scénario alternatif | SA1- Après 3 : le login et/ou le mot de passe sont incorrectes,   1. Le système envoie un message d’erreur.   L’enchainement va reprendre au point 1 |
| Postcondition | Authentification réussie |

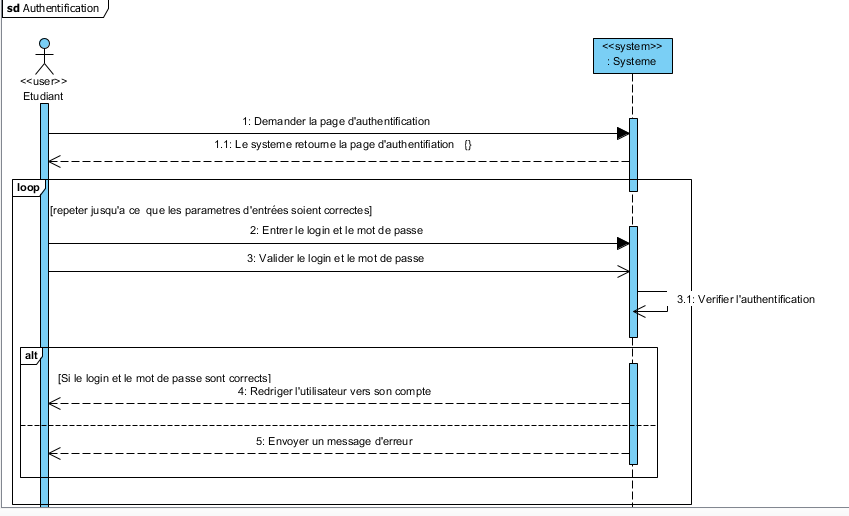


Figure 8 Diagramme de séquence du cas « S’authentifier »

* **Cas « Consulter Emploi du temps »**

|  |  |
| --- | --- |
| Cas | Consulter Emploi du temps |
| Objectif | Permettre à un étudiant de visualiser son emploi du temps via l’outil VT |
| Acteur(s) | Etudiant |
| Précondition | S’authentifier |
| Scénario nominal | 1. Demander la page d’affichage d’emploi du temps 2. Le système renvoie la page vt.esp.sn 3. Fournir au système son identifiant 4. Le système renvoie l’emploi du temps à travers vt.esp.sn |
| Scénario alternatif | SA1- Après 3 : identifiant saisi incorrect   1. Le système envoie un message d’erreur.   L’enchainement va reprendre au point 2 |
| Postcondition | Emploi du temps consulté avec succès |

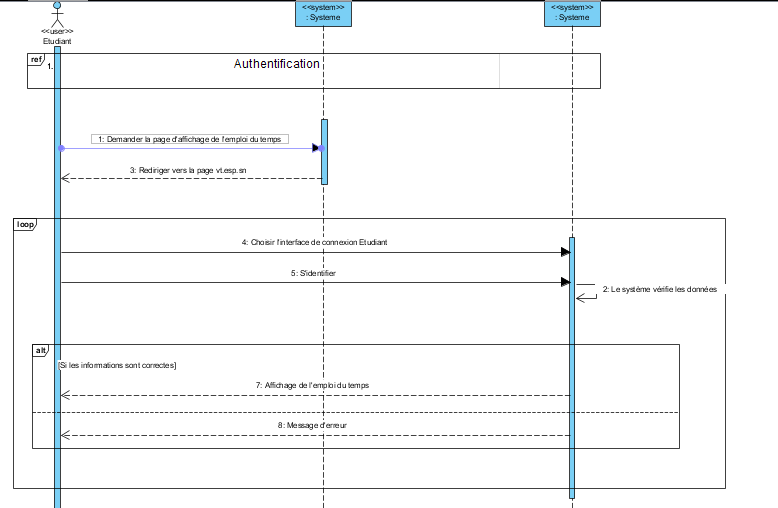
****

Figure 9 Diagramme de séquence du cas « Consulter emploi du temps »

**Cas « Consulter notes »**

|  |  |
| --- | --- |
| Cas | Consulter notes |
| Objectif | Permettre à l’étudiant de voir l’ensemble de ses notes disponibles |
| Acteur(s) | Etudiant |
| Précondition | S’authentifier |
| Scénario nominal | 1. Demander la page de consultation des notes 2. Vérification du système 3. Le système affiche les notes disponibles |
| Scénario alternatif | SA1- Après 2 : Aucune note n’est disponible pour le moment  3. Le système envoie un message d’erreur.  L’enchainement va reprendre au point 1 |
| Postcondition | Consultation de notes réussie |

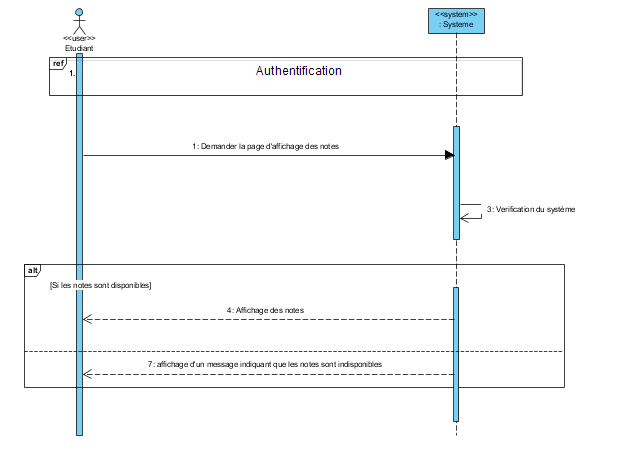
****

Figure 10 Diagramme de séquence du cas « Consulter notes »

**Cas « Enregistrer note »**

|  |  |
| --- | --- |
| Cas | Enregistrer note |
| Objectif | Permettre à l’enseignant d’enregistrer les notes des élèves |
| Acteur(s) | Enseignant |
| Précondition | S’authentifier |
| Scénario nominal | 1- Demander la page pour enregistrer une note  2- Le système renvoie la page  3-Fournir les informations : note, CC ou DS, prénom et nom de l’étudiant  4- Le système vérifie les données  5- Le système enregistre les données |
| Scénario alternatif | SA1 : Après 4 : si les informations sont incorrectes 5- Le système envoie un message d’erreur L’enchainement reprend à l’étape 2 |
| Postcondition | Enregistrement réussie |

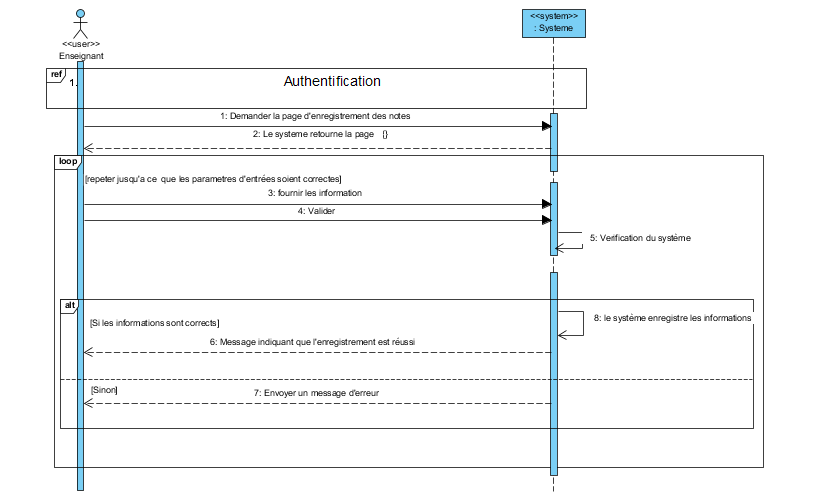


Figure 11 Diagramme de séquence du cas « Enregistrer notes »

**Cas « Ajouter enseignant »**

|  |  |
| --- | --- |
| Cas | Ajouter enseignant |
| Objectif | Permettre à la secrétaire d’ajouter un enseignant dans le système |
| Acteur(s) | Secrétaire |
| Précondition | S’authentifier |
| Scénario nominal | 1. Importer le fichier contenant les informations de l’enseignant 2. Valider 3. Vérification du système 4. Enregistrement des informations |
| Scénario alternatif | SA1 : Après 3 : si les informations sont incorrectes 5- Le système envoie un message d’erreur L’enchainement reprend à l’étape 1 |
| Postcondition | Enregistrement réussie |

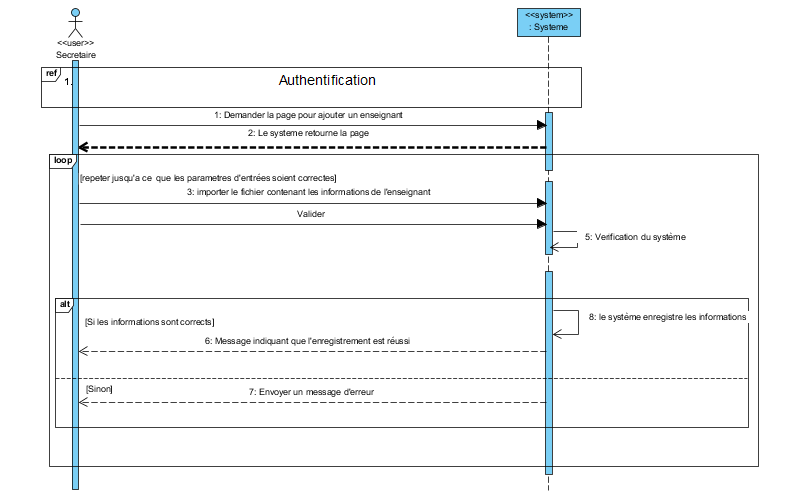
****

Figure 12 Diagramme de séquence du cas « Ajouter enseignant »

1. **Diagrammes de classes**

Ce diagramme illustré ci-dessous représente notre diagramme de classes. Ce diagramme, à première vue un peu complexe, est composé de 17 classes au total dont 2 classes d’énumérations et 2 classes d’association.

Les 13 autres classes sont modélisées de façon à interagir entre elles. Les paragraphes qui suivent constitue une description détaillée de notre diagramme de classes représentée un plus bas.

* Un étudiant fréquente une classe et est situé à un niveau précis de sa formation. Il peut s’absenter sur plusieurs séances et chaque absence est identifiée de manière unique. Il est évalué sur plusieurs éléments constitutifs (EC) d’une unité d’enseignement et chaque évaluation a une identification qui lui est propre.
* Un enseignant peut faire cours dans plusieurs classes et dans une classe peuvent intervenir plusieurs enseignants.
* La secrétaire peut publier plusieurs informations et est rattachée qu’à un seul département. Une information n’est publiée que par un secrétaire.
* Un EC peut se lier à plusieurs séances de cours et n’est inclus que dans une UE.
* Une séance de cours est liée à un et un seul élément constitutif. Durant une séance peuvent s’absenter plusieurs étudiants.
* Un semestre concerne plusieurs EC.
* Une unité d’enseignement inclut plusieurs éléments constitutifs et n’est rattaché qu’à un seul niveau de formation.
* Un niveau de formation peut être composé de plusieurs classes et dans un niveau sont situés plusieurs étudiants.
* Une classe est fréquentée par plusieurs étudiants et n’appartient qu’à un seul département. A un département appartiennent plusieurs classes.
* Les informations publiées peuvent être d’ordre d’actualité pédagogique ou de simple note de service.
* Le sexe de l’étudiant est soit du genre masculin, soit du genre féminin.

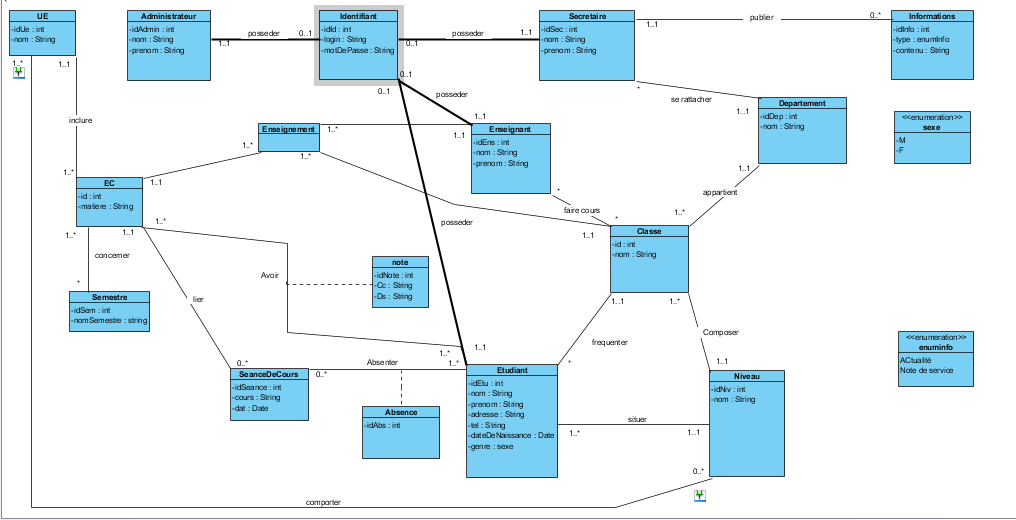


Figure 13 Diagramme de classes

1. **CONCLUSION**

Dans ce précédent chapitre, nous nous sommes intéressés à la phase d’analyse et de spécifications des besoins. Cette phase, bien qu’essentielle, constitue l’avant-garde de la phase de conception et de mise en œuvre du projet.

**CHAPITRE**

**3**

**CONCEPTION ET MISE EN ŒUVRE DE LA SOLUTION**

Dans ce présent chapitre, nous allons nous intéresser à la phase de conception et de mise en œuvre de la solution qui s’agit ici de l’application proprement dite. Cette phase est la phase ultime de notre projet et concrétise l’approche théorique réalisée jusqu’ici.

1. **CONCEPTION**

**I.1. Architecture 3 tiers**

Il s’agit d’un modèle logique d’architecture applicative qui vise à modéliser une application comme étant un empilement de trois (3) couches dont les rôles sont clairement définis :

* La **présentation** des données (premier niveau), correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ou l’interface homme-machine ;
* Le **traitement** métier des données (deuxième niveau), correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative, géré par l’application ;
* L'**accès aux données** [persistantes](https://fr.wikipedia.org/wiki/Persistance_(informatique)) (troisième niveau), correspondant aux données qui sont destinées à être conservées sur la durée, dans les serveurs de bases de données.

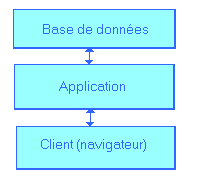


Figure 14 Illustration des 3 couches de l’architecture 3-tiers

Dans cette architecture, les couches communiquent selon un modèle d’échange et chaque couche propose un ensemble de services. Les services d’une couche sont à la disposition de la couche qui lui est directement supérieure. Il n’est pas permis dans ce modèle d’architecture qu’une couche utilise les services autres que celle de la couche qui lui est directement inférieure.

De plus, cette architecture permet une certaine indépendance entre les trois niveaux qui la composent.

L'architecture trois tiers a pour objectif de répondre aux préoccupations suivantes :

* L'allègement du poste de travail client ;
* La prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.) ;
* L'introduction de clients dits « légers » (plus liée aux technologies [Intranet](https://fr.wikipedia.org/wiki/Intranet)/[HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language) qu'à l'architecture trois tiers proprement dite) ;
* L'amélioration de la sécurité des données, en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche d'accès aux données ;
* La rupture du lien de propriété exclusive entre application et données. Dans ce modèle, la base de données peut être plus facilement normalisée et intégrée à un [entrepôt de données](https://fr.wikipedia.org/wiki/Entrep%C3%B4t_de_donn%C3%A9es) ;
* Une meilleure répartition de la charge entre différents [serveurs d'applications](https://fr.wikipedia.org/wiki/Serveur_d%27applications).

Une application web possède souvent une architecture trois tiers, le premier niveau étant représenté par le poste de l’utilisateur, le deuxième niveau par l’application web et le troisième par le système de gestion de base de données.

Notre choix porté sur l’architecture 3 tiers est une conséquence des avantages qu’offre cette architecture notamment une indépendance entre niveaux et une meilleure répartition des charges.

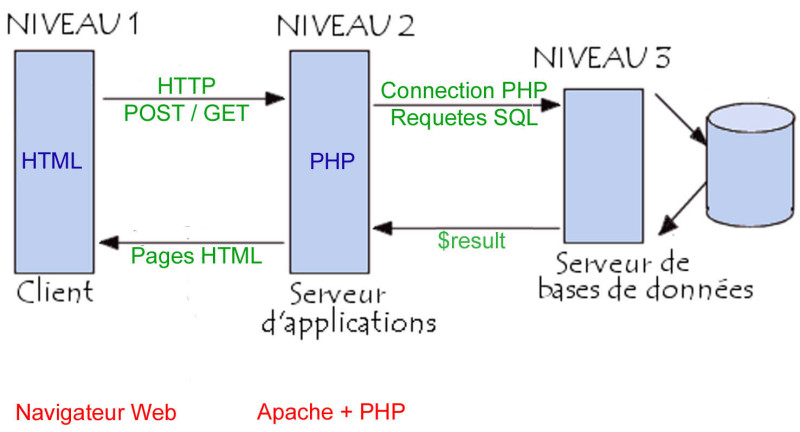


Figure 15 Représentation d’une application web avec l’architecture 3 tiers

**I.2. Langages et technologies utilisés**

Comme précédemment énoncée, la solution à la problématique posée est ici réalisée sous forme d’application web. Pourquoi avoir choisi une application web ? En effet, les applications web sont à moindre coût, rapidement accessible, compatible à tous les systèmes d’exploitation, accessible de partout, et proposant aussi une sécurité des données.

Comme pour tout autre type d’application, les applications web sont développées au moyen des langages de développement. Un langage de développement est une notation conventionnelle destinée à formuler des algorithmes et à produire des programmes informatiques qui les appliquent. Chaque langage possède d’un alphabet, d’un vocabulaire, et de ses propres règles.

Dans ce contexte, nous avons utilisé des langages de programmation web que énoncerons ci-après.

* **HTML et CSS**



L’**HyperText Markup Langage**, abrégé HTML, est un langage de balisage permettant de représenter des pages web. Il permet d’écrire de l’hypertexte d’où son nom. HTML permet de structurer sémantiquement et logiquement la mise en forme des pages et d’inclure des ressources multimédias. Bref, avec HTML, on conçoit des pages conformes aux normes du web.

Les feuilles de styles en cascades ou CSS (de l’anglais **Cascading Style Sheets**) forment un langage qui décrit la présentation des documents HTML et XML. Les standards du CSS sont publiés par le W3C (World Wide Web Consortium). Les feuilles de styles en cascades gèrent l’apparence, la décoration, la couleur, la taille du texte, etc.

En somme HTML gère le fond et CSS gère la forme d’une page web.

* **JavaScript**

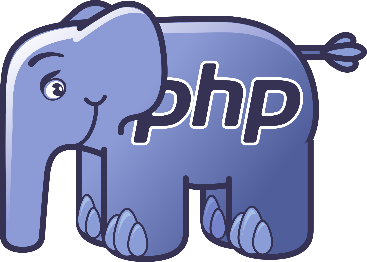


**JavaScript** est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) de [scripts](https://fr.wikipedia.org/wiki/Langage_de_script) principalement employé dans les [pages web](https://fr.wikipedia.org/wiki/Pages_web) interactives mais aussi pour les serveurs[2](https://fr.wikipedia.org/wiki/JavaScript#cite_note-2) avec l'utilisation (par exemple) de [Node.js](https://fr.wikipedia.org/wiki/Node.js)[3](https://fr.wikipedia.org/wiki/JavaScript#cite_note-3). C'est un langage [orienté objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) à [prototype](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_prototype), c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des [objets](https://fr.wikipedia.org/wiki/Objet_(informatique)) qui ne sont pas des [instances](https://fr.wikipedia.org/wiki/Instance_(programmation)) de [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)), mais qui sont chacun équipés de [constructeurs](https://fr.wikipedia.org/wiki/Constructeur_(programmation_informatique)) permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets [héritiers](https://fr.wikipedia.org/wiki/H%C3%A9ritage_(informatique)) personnalisés. En outre, les [fonctions](https://fr.wikipedia.org/wiki/Fonction_informatique) sont des [objets de première classe](https://fr.wikipedia.org/wiki/Objet_de_premi%C3%A8re_classe). Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances [npm](https://fr.wikipedia.org/wiki/Npm" \o "Npm), avec environs 500 000 paquets en août 2017[4](https://fr.wikipedia.org/wiki/JavaScript#cite_note-4).

JavaScript a été créé en 1995 par [Brendan Eich](https://fr.wikipedia.org/wiki/Brendan_Eich). Il a été standardisé sous le nom d'[ECMAScript](https://fr.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) en juin 1997 par [Ecma International](https://fr.wikipedia.org/wiki/Ecma_International" \o "Ecma International) dans le standard ECMA-262. Le standard ECMA-262 en est actuellement à sa 8e édition. JavaScript n'est depuis qu'une implémentation d'ECMAScript, celle mise en œuvre par la fondation [Mozilla](https://fr.wikipedia.org/wiki/Mozilla_Foundation). L'implémentation d'ECMAScript par [Microsoft](https://fr.wikipedia.org/wiki/Microsoft)(dans [Internet Explorer](https://fr.wikipedia.org/wiki/Internet_Explorer) jusqu'à sa version 9) se nomme [JScript](https://fr.wikipedia.org/wiki/JScript" \o "JScript), tandis que celle d'[Adobe Systems](https://fr.wikipedia.org/wiki/Adobe_Systems) se nomme [ActionScript](https://fr.wikipedia.org/wiki/ActionScript" \o "ActionScript).

De plus, le JavaScript est très avantageux car une très grande majorité des visiteurs l’acceptent. Il est relativement simple et facile à apprendre. Accompagné d’une certaine vitesse, le JavaScript est un langage versatile. JavaScript constitue ainsi une technologie vraiment incontournable.

* **PHP**



PHP est un langage interprété (un langage de script) exécuté du côté serveur (comme les scripts CGI, ASP, etc.) et non du côté client (un script écrit en JavaScript ou une applet Java s'exécute sur votre ordinateur...). La syntaxe du langage provient de celles du langage C, du Perl et de Java.

PHP permet la création de sites web dynamiques mais aussi des applications web. Sa version actuelle est la version 7.1.8.

Présent sur toutes les plateformes de système d’exploitation, PHP permet l’interaction avec les serveurs de base de données.

* **MYSQL**



MySQL est un système de gestion de bases de données relationnelles. Avec Oracle, Informix et Microsoft SQL Server, il fait partie des logiciels de bases de données les plus utilisés au monde.

MySQL est produit sous double licence : une licence gratuite et une licence payante. Il est multi-thread (plusieurs exécutions simultanées) et multi-utilisateur. Sa facilité d’utilisation explique le fait qu’il soit prisé des étudiants et autres apprenants en SGBD.

* **GITHUB**

GitHub est un service d’hébergement web (dépôt de codes) et de gestion de développement de d’applications utilisant le logiciel de gestion de versions Git. GitHub offre une panoplie d’avantages :

* La gestion des branches : GitHub permet de travailler sur plusieurs projets en parallèle.
* Une interface console facile d’utilisation.
* Les algorithmes de fusion de codes (merge) qui fusionnent intelligemment les lignes d’un fichier modifié simultanément par plusieurs utilisateurs.
* La rapidité de mise à jour des données d’un dépôt de code.

GitHub constitue alors une bonne plateforme de partage pour les développeurs web.

**I.3. Déploiement de la base de données**



Figure 16 Base de données de l’application

1. **IMPLEMENTATION DE L’APPLICATION**

**II.1. Environnement de développement**

* **WampServer**



WampServer est une plate-forme de développement Web sous Windows. Il vous permet de développer des applications Web dynamiques à l'aide du serveur http Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin et SQLite Manager pour gérer plus facilement vos bases de données.

Son installation est très facile et son utilisation très intuitive permet de le configurer très rapidement (sans toucher aux fichiers de configuration). WampServer vous permet de reproduire fidèlement votre serveur de production. Une fois la base installée, vous avez la possibilité d'ajouter autant de versions d’Apache, MySQL et PHP que vous le souhaitez.

* Apache : Il s’agit d’un serveur http ou serveur web permettant à des clients d’accéder à des pages web, c’est à dire en réalité des fichiers au format HTML à partir d’un navigateur (aussi appelé browser) installé sur leur ordinateur distant. C’est le serveur le plus répandu dans le web.
* MySQL : Le plus populaire des serveurs de bases de données SQL Open Source, MySQL est un système de gestion de bases de données (SGBD) robuste et rapide. Il s’exécute sur un serveur sous forme de service ou démon. Il fonctionne sur de nombreuses plateformes, disposant d’un système d’allocation de mémoire très rapide grâce à l’exploitation des threads. Ce SGBD offre des jointures très rapides, exploitant un système de jointures multiples en une seule passe optimisée. Le serveur de bases de données MySQL est très rapide, fiable et facile à utiliser.
* **Sublime Text**



**Sublime Text** est un [éditeur de texte g](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte)énérique codé en [C++ e](https://fr.wikipedia.org/wiki/C%2B%2B)t [Python,](https://fr.wikipedia.org/wiki/Python_(langage)) disponible sur [Windows,](https://fr.wikipedia.org/wiki/Windows) [Mac e](https://fr.wikipedia.org/wiki/Mac_OS)t [Linux.](https://fr.wikipedia.org/wiki/Linux) Le logiciel a été conçu tout d'abord comme une extension pour [Vim,](https://fr.wikipedia.org/wiki/Vim) riche en fonctionnalités. Depuis la version 2.0, sortie le [26 juin 2012,](https://fr.wikipedia.org/wiki/26_juin) l'éditeur prend en charge quarante-quatre langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares. Sublime Text intègre la plupart des fonctionnalités de base d'un [éditeur de texte,](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) dont la [coloration syntaxique p](https://fr.wikipedia.org/wiki/Coloration_syntaxique)ersonnalisable.

* **VISUAL PARADIGM (community edition)**



**Visual Paradigm** est unemultiplateforme riche en fonctionnalités puissantes qui dispose de tous les diagrammes UML et des outils DRE essentiellement dans le système et la conception de base de données. Visual Paradigm Community Edition fournit un environnement de modélisation visuelle plus facile à utiliser et intuitif avec un riche ensemble d'[importation et d'exportation](https://www.visual-paradigm.com/features/output-export-import/) des capacités telles que XMI, XML, PDF, JPG.

**II.2. Réalisation de la plateforme**

La réalisation de la plateforme est l’objectif visé depuis la formulation du sujet. La solution modélisée a été implémentée en application web. Cette application web est simple, explicite et facile à utiliser.

Des captures de l’application ont été prises en guise d’illustrations.

La première page du site est la page d’authentification de l’application. Dans cette page on fournit à travers un formulaire les identifiants de connexions. Si ces identifiants sont authentiques, on accède à son compte, sinon un message d’erreur vous est envoyé et vous reprenez l’authentification.



Figure 17 Page d’authentification du site



Figure 18 Page d’authentification du site

Ces pages qui suivent montrent les différentes fonctionnalités offertes à l’étudiant. Il s’agit entre autres de la consultation de notes, de l’emploi du temps, des absences et des informations.



Figure 19 Page de consultation des notes

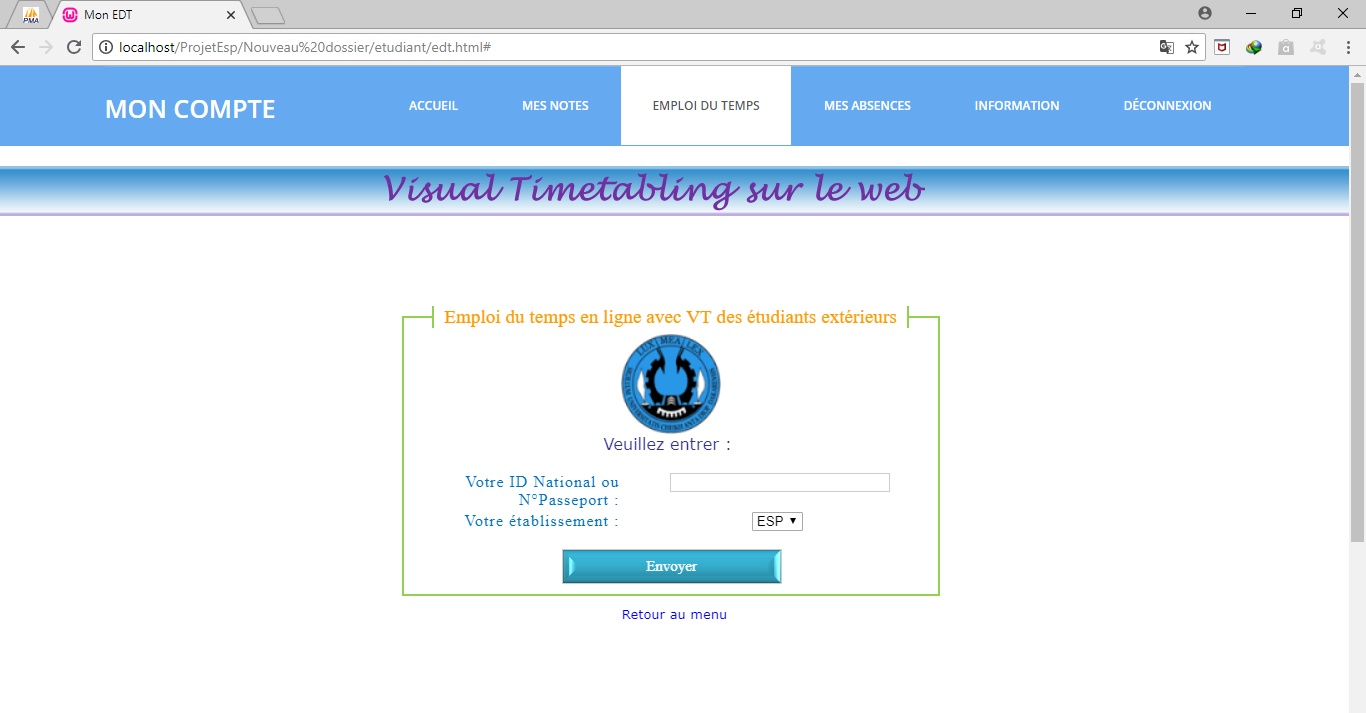
****

Figure 20 Page de consultation de l’emploi du temps

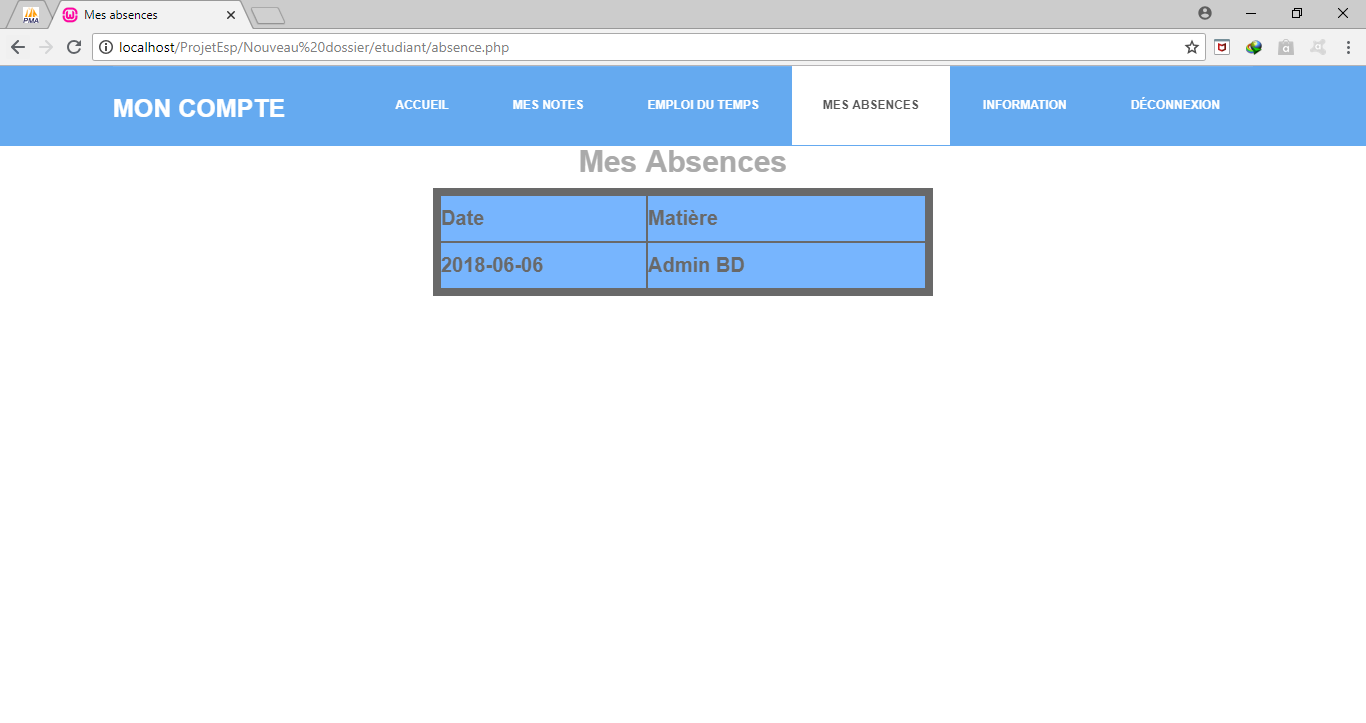


Figure 21 Page de consultation des absences

L’information est ici disponible en version PDF téléchargeable.

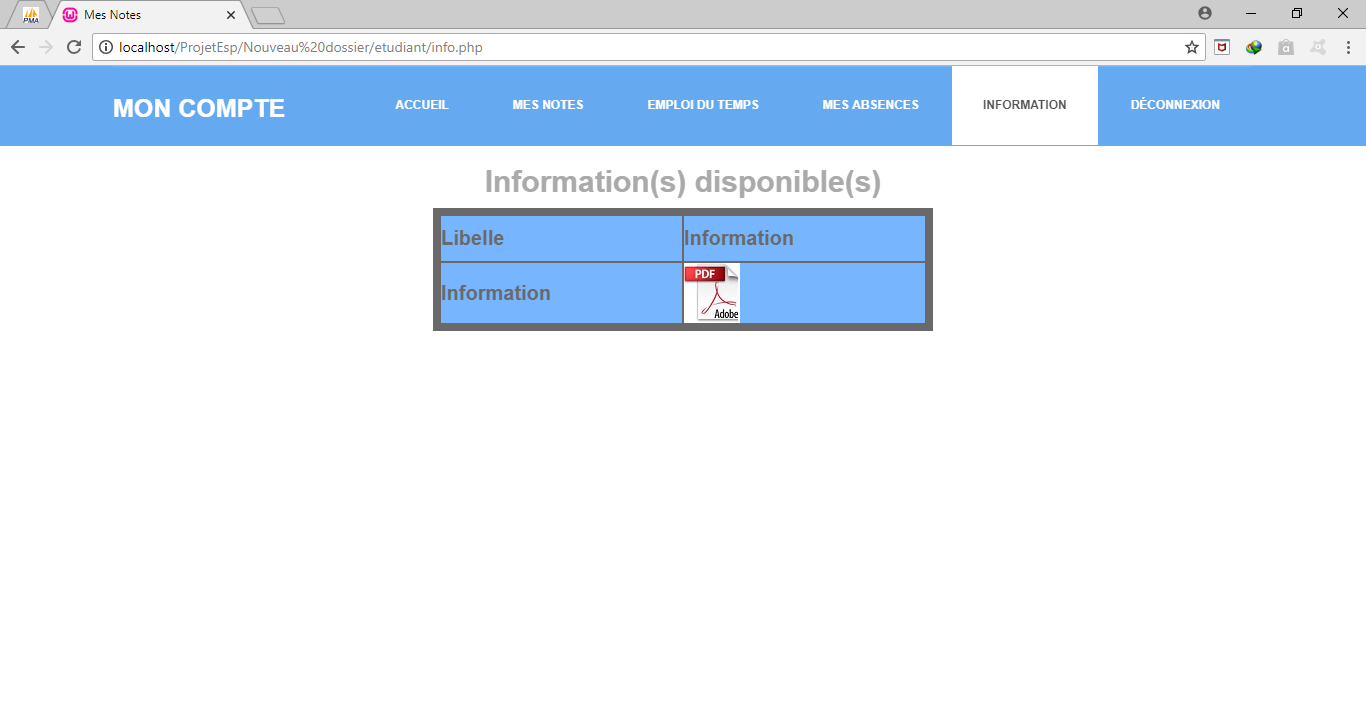


Figure 22 Page de consultation de l’information

Les captures suivantes montrent quelques services offerts par l’application à l’enseignant.

Après avoir choisi un élève, on enregistre son absence.

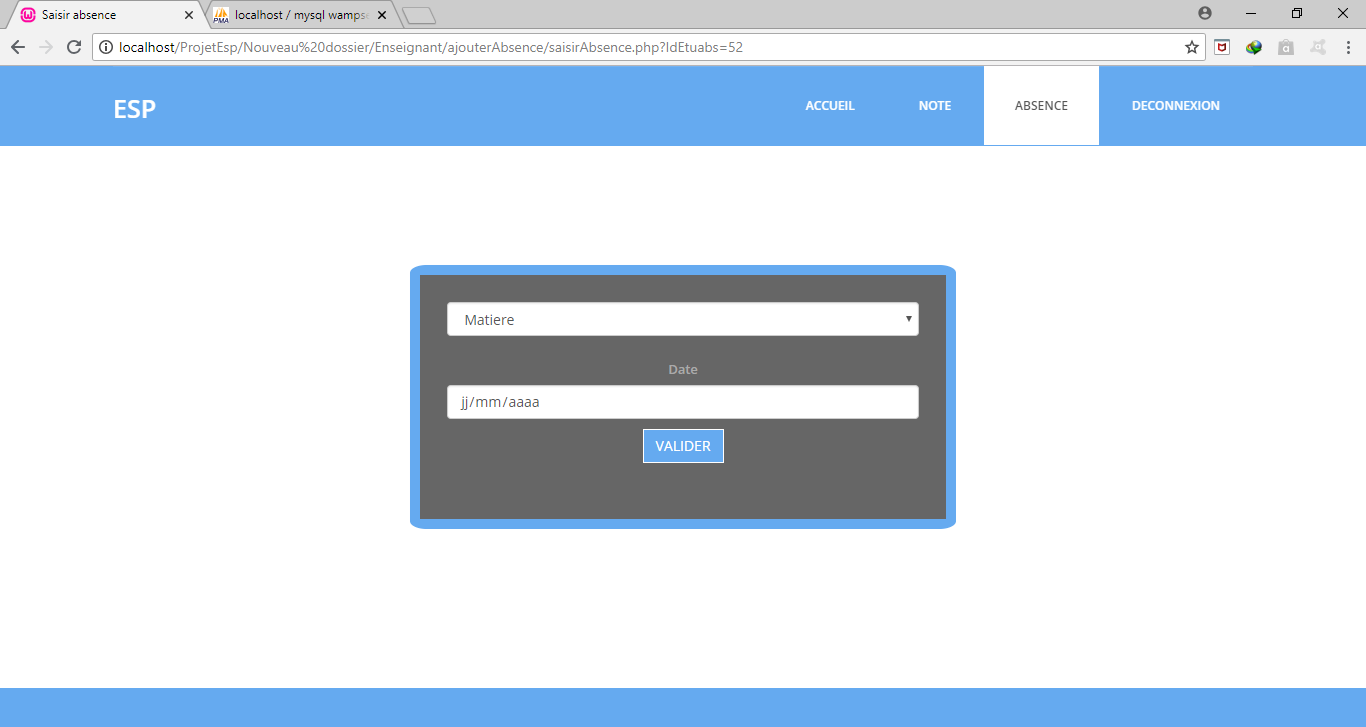


Figure 23 Page de saisie d’absence (Par l’enseignant)

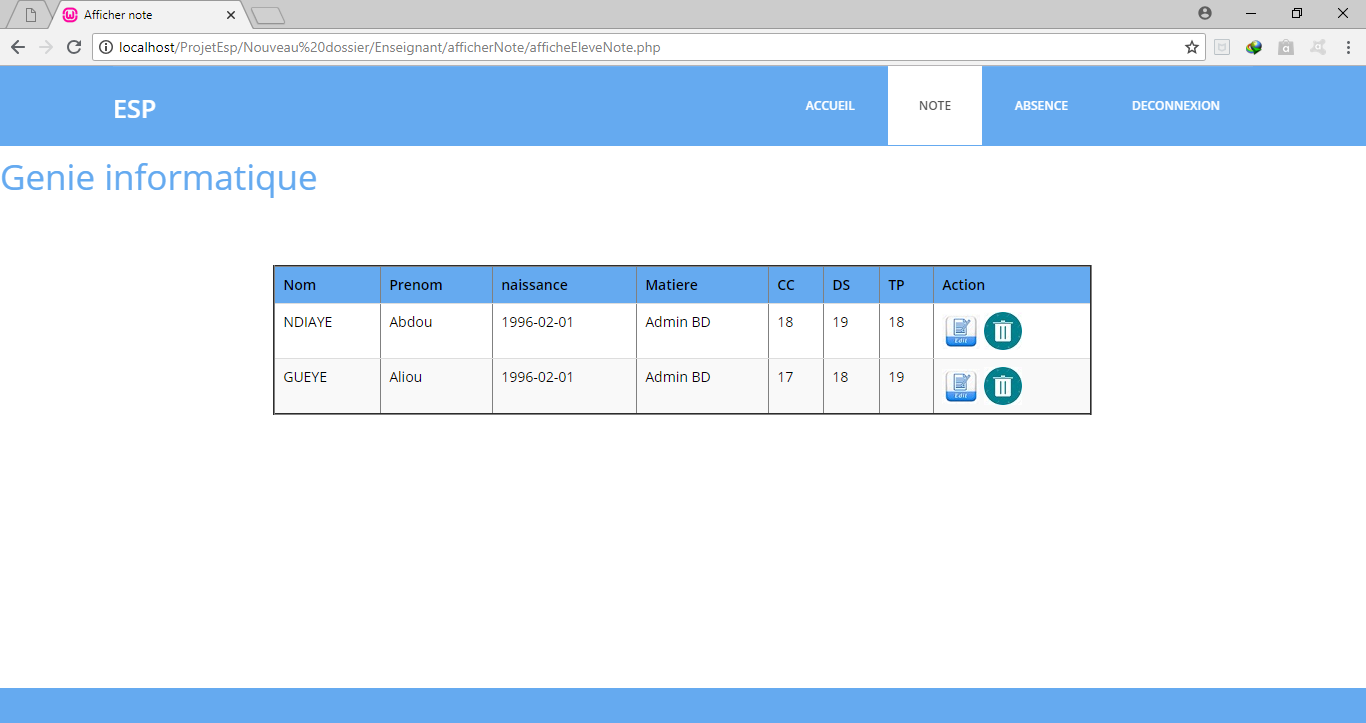


Figure 24 Page de mise à jour des notes des étudiants

Ces captures suivantes sont celles de quelques fonctionnalités de la secrétaire.

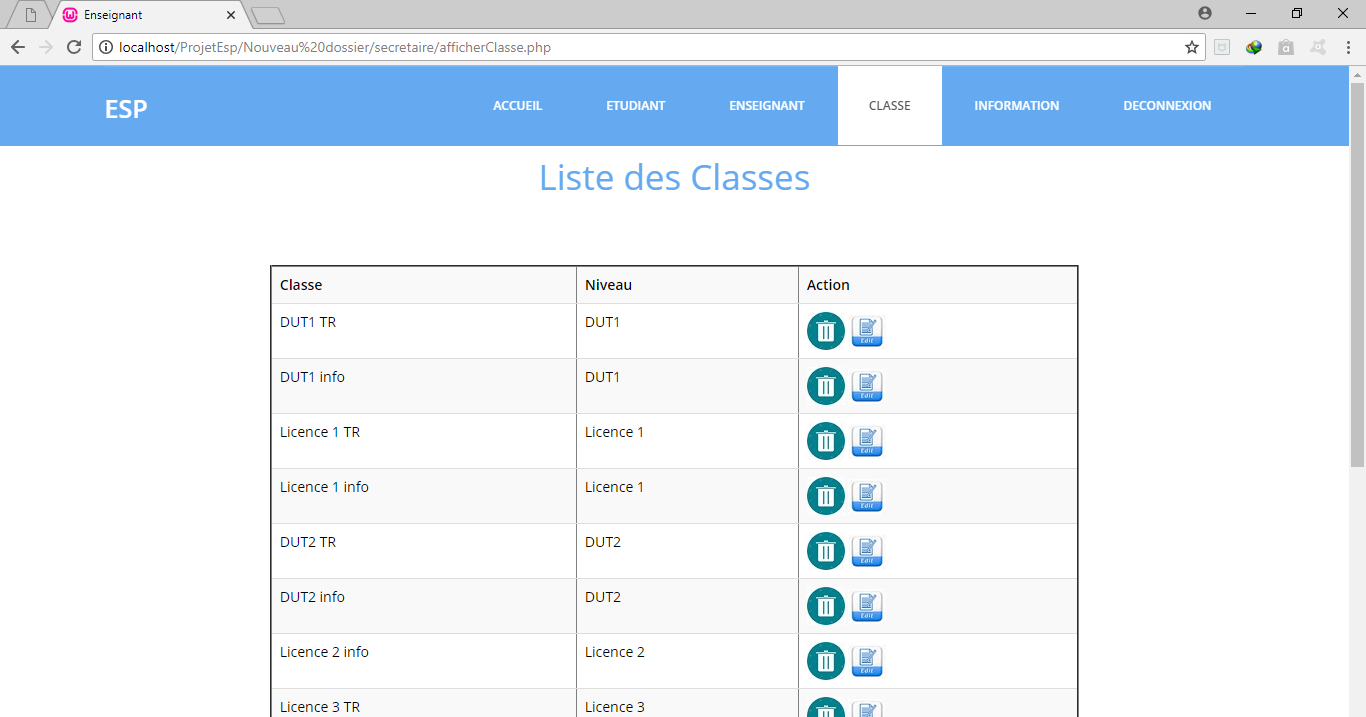


Figure 25 Page de mise à jour des classes

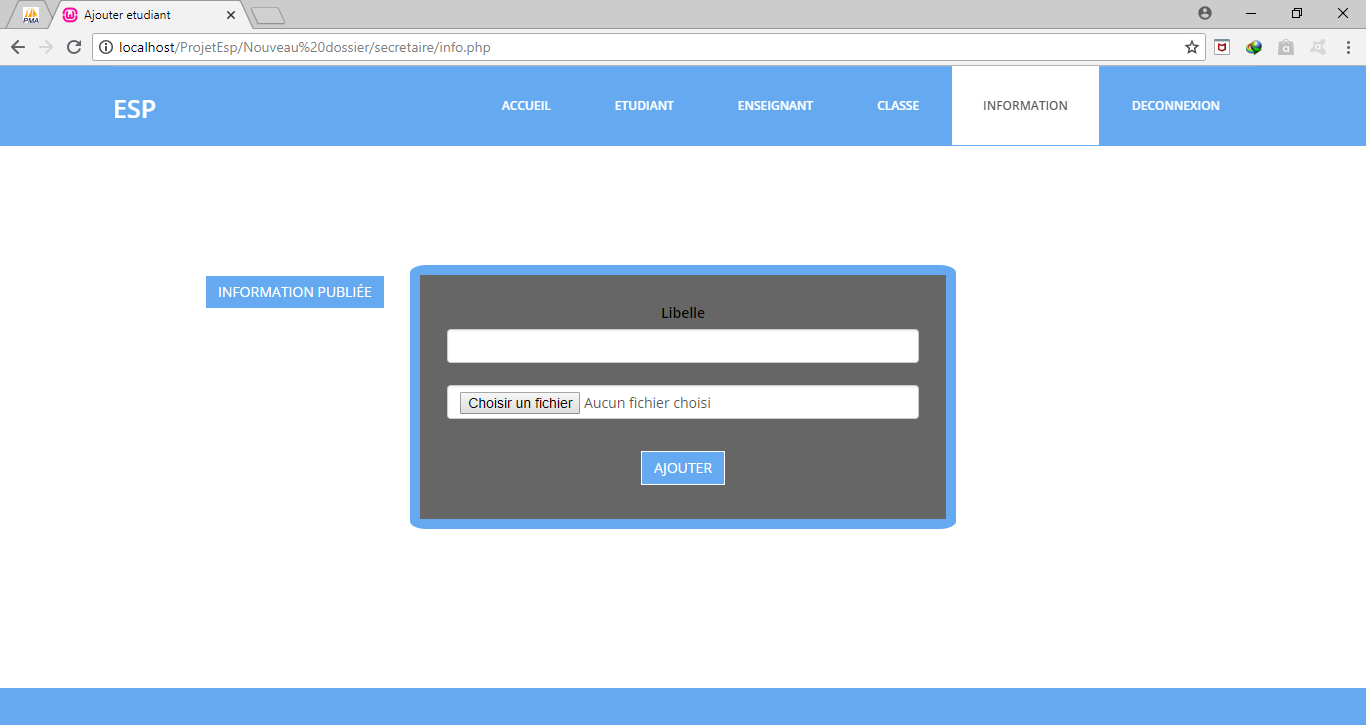


Figure 26 Page de publication de l’information

Enfin, nous terminons par quelques captures des fonctionnalités de l’administrateur.

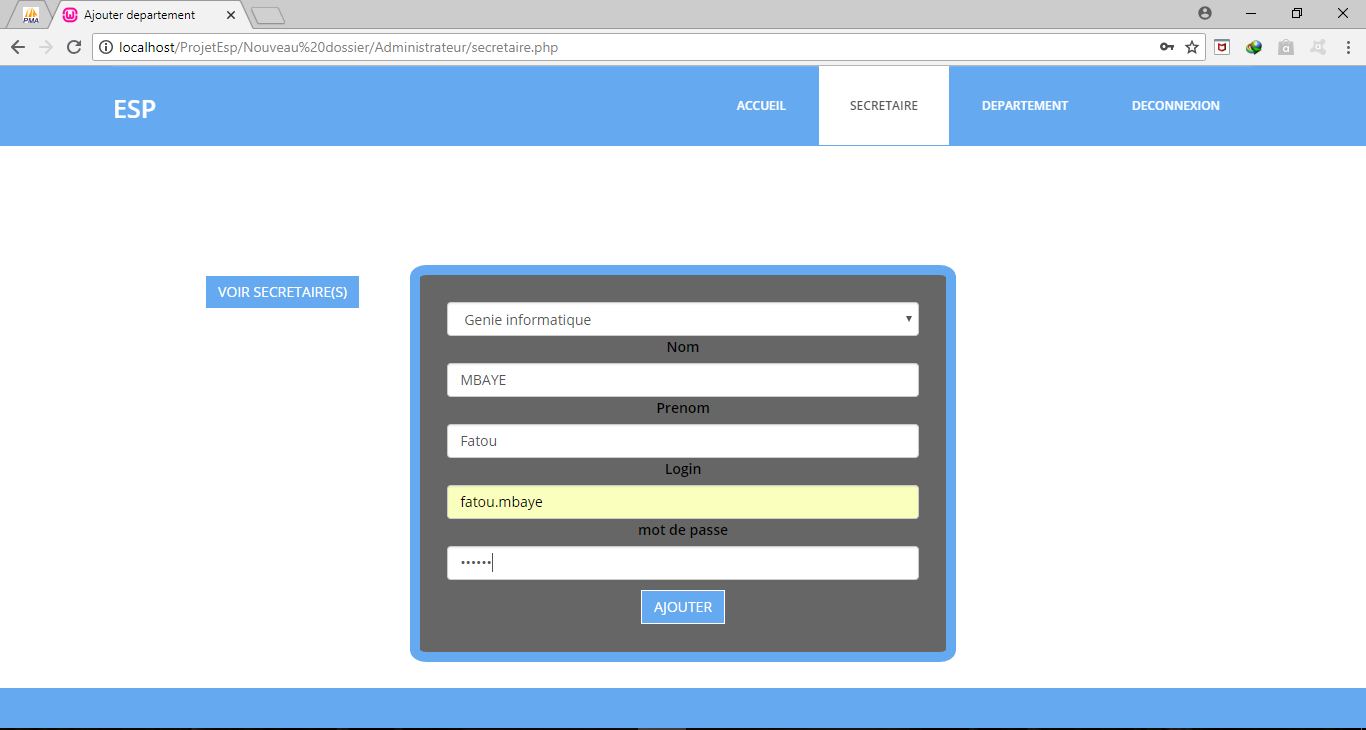


Figure 27 Page d’ajout d’une secrétaire

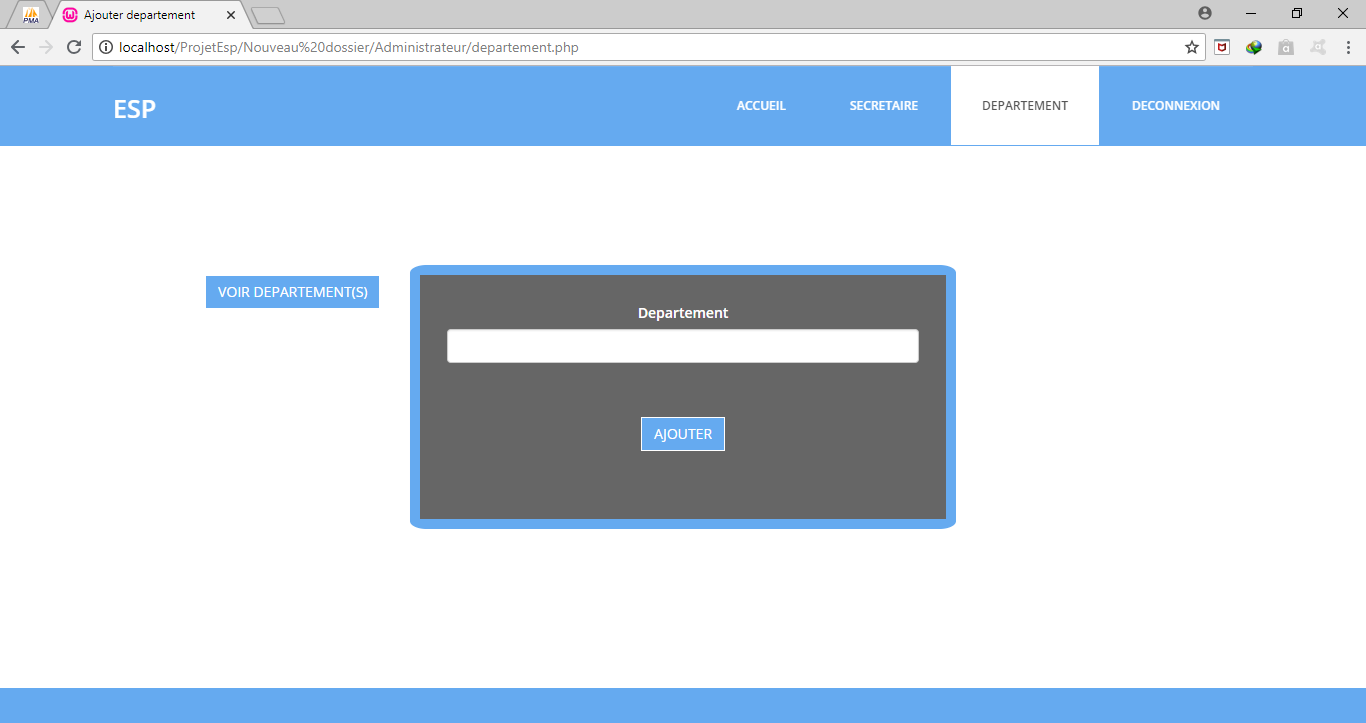


Figure 28 Page d’ajout d’une secrétaire



Figure 29 Page de listage des secrétaires

**CONCLUSION**

Le sujet soumis à notre étude de manière générale consistait à permettre un accès plus facile à l’information pour les étudiants de l’Ecole Supérieure Polytechnique. Etant donné que le site de l’ESP est de portée plus large que les outils de gestions des différents départements, il a été notre cible par une extension de ses fonctionnalités à travers la création d’espaces étudiant, enseignant, secrétaire et administrateur. La solution proposée, à la suite d’une analyse et une modélisation des acteurs et besoins, a été implémentée en application web avec une architecture 3 tiers. Les phases de développement du projet ont été axées sur la méthode du cycle en V qui minimise les retours aux étapes précédentes. Diverses technologies ont été utilisées parmi lesquelles GitHub qui aura été très utile pour le travail de l’équipe du projet à distance.

En termes de gains personnels, ce projet nous a permis d’avoir un aperçu beaucoup plus pratique que théorique des notions informatiques reçues en classe. Il nous a aussi permis de comprendre le bon fonctionnement d’un projet informatique en entreprise mais surtout nous a insufflé encore un peu plus du sens de la responsabilité.