|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

## Projet du Fin de Module

## Pour l’obtention de la

## Licence d’ingénierie et conception des systèmes

« Gestion d’Emploi du Temps »

Réalisée par  : Hossam Eddine ABOUELQASSIM

Encadrée par : Walid Bouarifi

Organisme : ENSA

Période : Du 01/10/2020 Au 30/10/2020

Remerciements

Au terme de mon projet de fin d’études, je tiens à adresser mes plus vifs remerciements

à toutes les personnes qui, de près ou de loin, ont contribué à l’aboutissement de ce travail dans les meilleures conditions.

Je remercie vivement les membres du jury que je remercie d’avoir accepté d’évaluer ce

projet.

Je m’adresse en premier lieu à mon encadrant Mr. Bouarifi Walid pour la qualité de

l’encadrement qu’il m’a assuré, pour sa disponibilité, ses conseils et remarques judicieuses

et surtout pour sa confiance et ses encouragements continuels qui m’ont permis d’avancer.

Enfin, nous sommes reconnaissants à tout le personnel et le corps professoral de l’ENSA et à tous ceux qui nous ont apporté de l’aide même minime.

*Résumé*

Le présent travail intitulé «Gestion d’Emploi du Temps» s'inscrit dans le cadre de projet de fin d'études en vue de l'obtention de diplôme de licence d’ingénierie et conception des systèmes d’information . Notre projet a pour objectif de mettre en place une application Gestion d’emploi du temps qui permettra de gérer plusieurs modules tels que la gestion des enseignants, des étudiants, des salles, des filières...

Il s’agit d’une application qui regroupe les activités de gestion d’emploi du temps dans les établissements universitaires. Elle se consacre à la conception et à l’implémentation d’un outil capable d’automatiser, ainsi de faciliter et d’optimiser cette tâche.

Cet outil permet la manipulation des données telle que l’ajout, la suppression, la modification, et la sélection des documents intégrés dans l’application Gestion d’Emploi du Temps.

**Mots clés :** HTML, CSS3, JavaScript, HTML5, PHP, MySQL, UML

**Sommaire**

[***Introduction Générale*** 8](#_Toc390005074)

**Chapitre I** [**Présentation et analyse du problème** 10](#_Toc390005076)

[Introduction 11](#_Toc390005077)

[I. Présentation de l’organisme d’accueil 11](#_Toc390005078)

[1. Présentation générale 11](#_Toc390005079)

[2. Présentation du projet 11](#_Toc390005080)

[II. Problématique 13](#_Toc390005081)

[1. Critique de l’existant 13](#_Toc390005082)

[2. Les solutions proposées 13](#_Toc390005083)

[3. Processus de développement 13](#_Toc390005084)

[III. Analyse de besoins et spécification 15](#_Toc390005085)

[1. Spécification des besoins fonctionnels 15](#_Toc390005086)

[2. Spécification des besoins non fonctionnels 16](#_Toc390005087)

[Conclusion 16](#_Toc390005088)

**Chapitre II**  [**Etude conceptuelle** 17](#_Toc390005090)

[Introduction 18](#_Toc390005091)

[I. Technique de modélisation 18](#_Toc390005092)

[1. Choix méthodologique et conceptuel 18](#_Toc390005093)

[2. Diagramme de cas d’utilisation 18](#_Toc390005094)

[3. Diagramme de classe 28](#_Toc390005095)

[4. Diagrammes de séquence 30](#_Toc390005096)

[Conclusion 32](#_Toc390005097)

**Chapitre III** [**Réalisation** 33](#_Toc390005099)

[Introduction 34](#_Toc390005100)

[I. Environnement de développement 34](#_Toc390005101)

[1. Environnement matériel 34](#_Toc390005102)

[2. Environnement logiciel 34](#_Toc390005103)

[3. Choix technique 35](#_Toc390005104)

[4. Choix du langage HTML5 35](#_Toc390005105)

[II. Aperçu sur le travail réalisé 35](#_Toc390005106)

[1. Interface d’accueil 36](#_Toc390005107)

[2. Interface d’authentification 36](#_Toc390005108)

[3. Interface d’ajouter un enseignant 37](#_Toc390005109)

[4. Interface de modifier un enseignant 39](#_Toc390005110)

[5. Interface de supprimer un enseignant 41](#_Toc390005111)

[8. Interface d’ajouter un groupe 49](#_Toc390005115)

[9. Interface de modifier un groupe 51](#_Toc390005116)

10- Interface d'afficher emplois du temps

[11. Chronogramme 53](#_Toc390005117)

[Conclusion 54](#_Toc390005118)

[***Conclusion Générale*** 55](#_Toc390005119)

[**Bibliographie & Netographie** 56](#_Toc390005120)

Liste des figures

Figure 1 : gestion classique d’emploi du temps12

F**igure 2 : Modèle par incrément**14

Figure 3 : Architecture technique proposée pour le problème des emplois du temps15

Figure 4 : Relation entre les acteurs19

**Figure 5 : Diagramme de cas d'utilisation «gérer enseignant»**21

**Figure 6 : Diagramme de cas d'utilisation «gérer étudiant »**22

**Figure 7 : Diagramme de cas d'utilisation «gérer matière»**23

**Figure 8 : Diagramme de cas d'utilisation «gérer locaux»**24

**Figure 9 : Diagramme de cas d'utilisation «gérer horaire»**25

**Figure 10: Diagramme de cas d'utilisation «gérer groupe»**28

**Figure 11 : Diagramme de classe**29

**Figure 12 : diagramme de séquence «ajouter notes »**30

**Figure 13 : diagramme de séquence «modifier notes »**31

**Figure 14 : interface d’accueil**36

**Figure 15 : interface d’authentification**37

**Figure 16 : interface 1 d’ajouter enseignant**38

**Figure 17 : interface 2 d’ajouter enseignant**38

**Figure 20 : interface 1 Edition enseignant**39

**Figure 21 : interface 2 Edition enseignant**40

**Figure 22 : interface 3 Edition enseignant**41

**Figure 23 : interface 1 de Suppression enseignant**42

**Figure 24 : interface 2 de Suppression enseignant**43

**Figure 34 : Interface1 Ajout groupe**50

**Figure 35 : Interface2 Ajout groupe**51

**Figure 36 : Interface 1 modifier groupe**52

**Figure 37 : Interface 2 modifier groupe**52

**Figure 38 : Interface 3 modifier groupe**53

**Figure 39 : Le chronogramme du travail**54

# *Introduction Générale*

### 

### De nos jours, les organisations connaissent de perpétuels changements (restructurations, évolutions de métiers, mutations technologiques…). D’autre part, des nombreuses universités migrent de plus en plus vers les systèmes informatiques centralisés au détriment de leurs systèmes de gestion de bases de données traditionnels. Le besoin en matière d’accès aux informations est devenu pressant pour permettre aux utilisateurs de gérer et de suivre tout ce qui rapporte à l’université et surtout la gestion des emplois du temps.

Ces systèmes d’informations se révèlent un vecteur de performance essentiel pour la productivité de l’université puisqu’ils sont des outils logiciels de pilotage de l’activité départementale et administrative, permettant à une université, et en particulier ENSA, de gérer et d’optimiser cette activité.

En effet, les activités administratives au sein d’une université sont diverses (gestion des emplois de temps, gestion des étudiants, gestion des salles,…) et gérés par plusieurs employés (chef des départements, enseignants, employé administratif…). Donc, il important de développer un système informatique centralisé pour que l’information soit accessible à tous.

Sur le marché de logiciels, ces genres de systèmes informatiques existent d’une manière indépendante, c'est-à-dire que pour chaque activité au sein de l’université un logiciel spécifique le gère. Ce qui amène à une redondance des données, difficultés à manipuler les informations et un travail répétitif qui mène à une perte de temps.

D’où l’importance de développer un système informatique qui assemble toutes les activités d’une université dans une seule plateforme. Plateforme centralisée, accessible à distance et qui rend l’information universitaire disponible.

C’est dans ce cadre que s’intègre notre projet de fin d’étude qui s’est déroulé au sein d’Ecole Nationale des Sciences Appliquées de Safi, établissement universitaire situé à Safi

De ce fait, notre objective visait à mettre en œuvre une solution logicielle intégrée et hautement paramétrable pour la gestion d’emploi du temps qui souscrit les différentes opérations demandées par l’Université.

Notre solution doit être puissante et souple à la fois, permettant d’organiser le travail.

Afin d’illustrer la démarche de notre travail, nous présentons dans ce qui suit l’organisation générale de ce rapport. En effet, le présent rapport s’articule autour de trois chapitres.

Le premier chapitre intitulé "Présentation et analyse du problème ", est réservé à l’étude de l’existant en cherchant les objectifs à développer, décrit les besoins de notre projet, qu’ils soient fonctionnels ou non fonctionnels.

Dans le deuxième chapitre, nous allons présenter la conception de l’application.

Dans le troisième chapitre, nous allons présenter la phase de l’implémentation de l’application en justifiant les différents choix techniques et technologiques utilisés au cours de cette phase avant de projeter quelques captures écran de l’application finale réalisée.

Nous terminons ce rapport par une conclusion générale.

# 

# 

*Chapitre I*

# Présentation et analyse du problème

# Introduction

La mise en place d’un logiciel, passe essentiellement par une étude de l’existant, qui offre la connaissance précise et rigoureuse du contexte sur lequel nous opérons et guide ainsi à une décision pour démarrer ou simplement marqueter un tel projet. C’est le sujet de ce chapitre qui présentera les principaux aspects techniques et normatifs sur lesquels reposera notre travail.

# Présentation de l’organisme d’accueil

## Présentation générale

1. Le réseau des **Écoles nationales des sciences appliquées** (**ENSA**) est un groupe d'écoles publiques [marocaines](https://fr.wikipedia.org/wiki/Maroc) et françaises délivrant des formations d'[ingénieur](https://fr.wikipedia.org/wiki/Ing%C3%A9nieur) d'État et ayant pour particularité commune de comporter un [cycle préparatoire intégré](https://fr.wikipedia.org/wiki/Cycle_pr%C3%A9paratoire_int%C3%A9gr%C3%A9).
2. Le réseau ENSA a connu pour la première fois un rassemblement les 6 et 7 mai 2011 à Agadir, au Forum ENSA Maroc[1](https://fr.wikipedia.org/wiki/%C3%89coles_nationales_des_sciences_appliqu%C3%A9es#cite_note-1) et en 2018 l'INSA Euro-Méditerranéen de Fès qui fait partie du réseau français INSA a rejoint le réseau des ENSA du Maroc avec un concours commun d'accès.

## Présentation du projet

### Cadre du projet

Ce projet s’inscrit dans le cadre de la réalisation de projet de fin d’études au sein de l’ENSA Safi et intitulé « Gestion d’Emplois du Temps ». Le but de ce travail est de développer un logiciel qui permet de garantir plus de fonctionnalités spécifiquement dédié aux établissements universitaires pour fournissant un moyen de recherche et de gestion plus efficace.

### Introduction à la Gestion d’Emplois du Temps

Chaque année, les responsables pédagogiques des universités et des Ecole Nationale des Sciences Appliquées  (ENSA) ont pour mission d’organiser les emplois du temps des différentes formations ou filières en essayant, au mieux, de satisfaire les contraintes «humaines» des enseignants et des étudiants, les contraintes pédagogiques imposées par la progression des enseignements et en tenant compte des contraintes « physiques » liées aux ressources matérielles (les salles, les équipements, etc.).

On constate, en effet, que lors de la création d’un emploi du temps, les responsables partent généralement d’un emploi du temps existant auxquels ils apportent des modifications. La méthode ne consiste donc pas à créer mais à adapter un emploi du temps. D’où la nécessité de disposer d’outils interactifs facilitant ces adaptations. Contrairement aux outils de génération automatique, les outils recherchés sont « centrés utilisateur ». Il convient de revenir à des outils totalement sous le contrôle de l’utilisateur, adaptés aux démarches de travail selon différents points de vue liés aussi bien à la conception, qu’à la visualisation (consultation et recherche) et la modification des emplois du temps. Dans cette optique, l’utilisateur redevient l’acteur central de la résolution du problème, et les outils s’articulent autour de son activité. Dans la suite, nous parlerons donc des acteurs (utilisateurs) de l’application.

L’outil recherché doit donc avant tout être interactif, souple, ouvert et doit présenter des qualités ergonomiques. Ces raisons nous ont motivés à réfléchir sur la conception d’un outil d’aide à la manipulation des emplois du temps suffisamment ouvert pour permettre de nombreuses extensions. Cette réflexion a abouti à la conception notre application de Gestion d’Emploi du Temps.

Cette application permet de gérer plusieurs modules. Parmi ces modules nous pouvons citer la gestion d’emploi du temps, des salles, des matières, des enseignants, des étudiants,...

Figure 1 : gestion classique d’emploi du temps

# Problématique

Dont le but de satisfaire les besoins de l’ENSA en développant un logiciel de qualité, Ce logiciel permettra de dégager les inconvénients du travail manuel et la propagation de temps.

## Critique de l’existant

Actuellement, la gestion des emplois de temps à l’ENSA de Safi se fait manuellement sur papier. Ensuite, les emplois de temps seront enregistrés dans un document Word. Ce qui rend ce travail pénible et difficile à maintenir.

A l’ENSA de Safi, où j’ai passé mon projet fin d’étude, ils existent plusieurs déficiences à la préparation de plusieurs tâches, tels que :

* La perte de temps lors de la création manuelle des emplois de temps.
* L’utilisation classique des feuilles.
* Génération des emplois de temps des groupes, des salles, et des enseignants se fait d’une manière manuelle (n’est pas automatique).
* Plusieurs difficultés dans l’opération de recherche des documents.

## Les solutions proposées

Pour résoudre les problèmes signalés précédemment, nous proposons de :

* Développer une application contenant tous les modules de gestion d’emploi du temps.

## Processus de développement

Nous avons choisi le modèle par incrément pour la conception et le développement de notre application puisque ce modèle propose un développement du logiciel par morceaux, lesquels sont livrés successivement au client, en venant se greffer à un noyau logiciel. Dans les modèles par incrément un seul ensemble de composants est développé à la fois : des incréments viennent s’intégrer à un noyau de logiciel développé au préalable.

Les avantages de ce type de modèle sont les suivants :

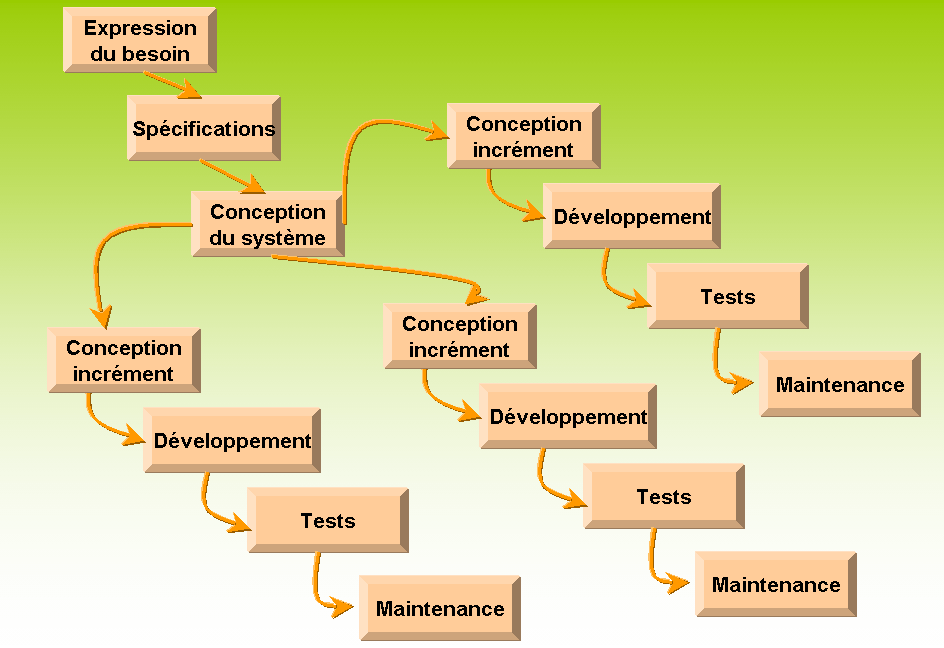
* Chaque développement est moins complexe.
* Les intégrations sont progressives.
* Il est ainsi possible de livrer et de mettre en service chaque incrément.
* Il permet un meilleur lissage du temps et de l’effort de développement grâce à la possibilité de recouvrement (parallélisassions) des différentes phases.

Les risques de ce type de modèle sont les suivants :

* Remettre en cause les incréments précédents ou pires le noyau.
* Ne pas pouvoir intégrer de nouveaux incréments.

Les noyaux, les incréments ainsi que leurs interactions doivent donc être spécifiés globalement, au début du projet. Les incréments doivent être aussi indépendants que possibles, fonctionnellement.

La figure 2 illustre le modèle de développement par incrément.



F**igure 2 : Modèle par incrément**

# Analyse de besoins et spécification

## Spécification des besoins fonctionnels

Dans cette section nous allons étudier les besoins fonctionnels, non fonctionnel et technique du notre système.

* L’administrateur peut :

Gérer les emplois du temps, les groupes, les étudiants, les enseignants, les matières, les locaux.

.

## Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le logiciel pour sa réalisation et son bon fonctionnement :

### BNF 1 Ergonomie et souplesse

### L'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.

### BNF 2 Rapidité

### L'application doit optimiser les traitements pour avoir un temps de génération de schéma raisonnable.

### BNF 3 Efficacité

### L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.

### BNF 4 Maintenabilité et capabilité

### Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins.

## Conclusion

Une étude des applications existantes concurrentes s'est avérée indispensable pour notre étude théorique du sujet. Lors de cette étude nous avons recouru à une comparaison entre quelques logiciels de gestion de note pour les universités présents en identifiant leurs limites et les objectifs de notre logiciel et ses apports.

Au niveau du chapitre suivant nous allons entamer notre étude conceptuelle.

*Chapitre II*

**Etude conceptuelle**

## Introduction

Après avoir défini la spécification du projet, la phase de conception vient pour mieux l’éclaircir. En effet, une phase de conception a pour objectif de permettre de formaliser les étapes préliminaires du développement d’un système afin de rendre ce dernier plus fidèle aux besoins des acteurs.

Elle permet de décrire de manière non ambigüe le fonctionnement désiré du système afin d’en faciliter la réalisation et la maintenance. Dans une première partie nous présenterons une conception globale de notre projet. Ensuite, dans la deuxième partie nous détaillerons la conception en utilisant les diagrammes UML appropriés.

# Technique de modélisation

## Choix méthodologique et conceptuel

Nous avons utilisé, pour la spécification et la conception de ce travail, la méthodologie UML (Unified Modeling Langage), qui est un langage de modélisation qui permet de décrire les besoins, de documenter les systèmes et d’esquisser les architectures logicielles. UML s’articule autour de neuf diagrammes différents, chacun d’eux étant dédié à la présentation des concepts particuliers d’un système logiciel.

Toutefois, nous ne présenterons que quelques diagrammes que nous avons jugés utiles et suffisants pour comprendre le projet à savoir les diagrammes de cas d’utilisations, les diagrammes de classes et les diagrammes de séquences.

## Diagramme de cas d’utilisation

Les cas d’utilisation permettent de traduire les spécifications fonctionnelles d’utilisation du système. Ils décrivent sous la forme d’actions et de réactions le comportement du système du point de vue d’un utilisateur et représentent un ensemble de séquences d’action réalisées par le système et produisant un résultat observable pour un acteur particulier.

Ce projet est reparti en 2 partie l’un que nous avons la nommé GNAG ce l’abréviation de Gestion de Note, d’Absence, et de Groupe, dans ce cas nous avons besoin de la gestion des enseignant, des étudiants, des locaux et des matières. Et l’autre partie était destinée à autre groupe qui comporte la gestion d’emploi de temps, et de rattrapage.

### Les acteurs de l’application

Un acteur est une entité externe qui agit sur le système. Il peut consulter ou gérer l’état du système selon son rôle.

Notre application a comme des acteurs :

* **Administrateur :** l’administrateur désigne le facteur humain responsable de l’administration de l’application.

### Relations entre cas d'utilisation

Deux types des relations standards entre cas d'utilisation sont proposés par UML :

• **«include» :** le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié,

• **«extend» :** le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.

### Diagramme des cas d’utilisation

Le diagramme de cas d’utilisation représente les acteurs, les cas d’utilisation et les relations entre eux.

* **Gérer enseignant**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer les enseignants.

**Enchaînement**

Le système affiche la page d’authentification.

L’utilisateur saisit son login et mot de passe et choisit le module prévue « Enseignants»

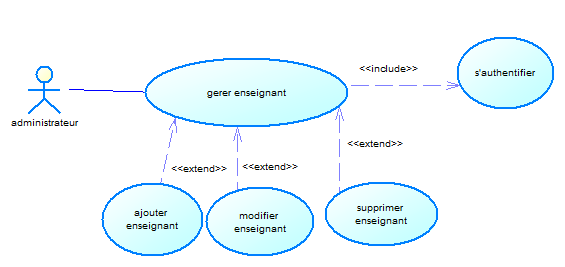
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’utilisateur fournit des paramètres d’identification inconnus, le système affiche un message d’erreur et demande de saisir un nouveau mot de passe.

La figure 5 présente le diagramme de cas d’utilisation « gérer enseignant »



**Figure 5 : Diagramme de cas d'utilisation «gérer enseignant»**

* **Gérer étudiant**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer les étudiants.

**Enchaînement**

Le système affiche la page d’authentification.

L’utilisateur saisit son login et mot de passe et choisit le module prévue « Etudiants »

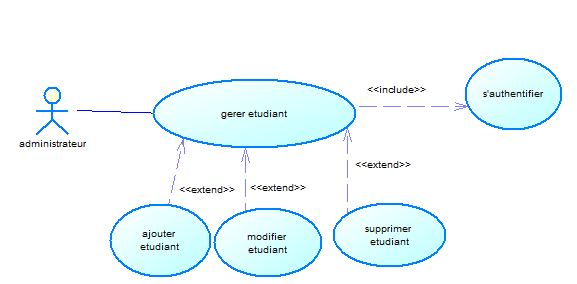
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’utilisateur fournit des paramètres d’identification inconnus, le système affiche un message d’erreur et demande de saisir un nouveau mot de passe.

La figure 6 présente le diagramme de cas d’utilisation « gérer étudiant »



**Figure 6 : Diagramme de cas d'utilisation « gérer étudiant »**

* **Gérer matières**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer les matières.

**Enchaînement**

Le système affiche la page d’authentification.

L’utilisateur saisit son login et mot de passe et choisit le module prévue « Matières »

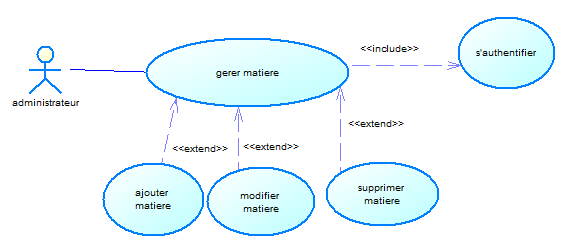
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’utilisateur fournit des paramètres d’identification inconnus, le système affiche un message d’erreur et demande de saisir un nouveau mot de passe.

La figure 7 présente le diagramme de cas d’utilisation « gérer matières »



**Figure 7 : Diagramme de cas d'utilisation « gérer matière »**

* **Gérer locaux**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer les locaux.

**Enchaînement**

Le système affiche la page d’authentification.

L’utilisateur saisit son login et mot de passe et choisit le module prévue « Locaux »

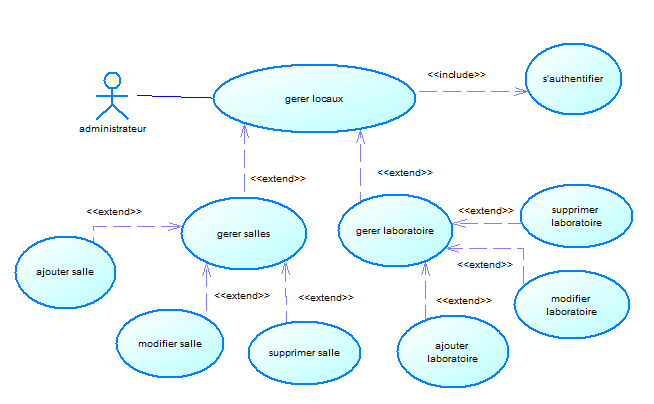
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’utilisateur fournit des paramètres d’identification inconnus, le système affiche un message d’erreur et demande de saisir un nouveau mot de passe.

La figure 8 présente le diagramme de cas d’utilisation « gérer locaux »



**Figure 8 : Diagramme de cas d'utilisation « gérer locaux»**

* **Gérer horaire**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer l’horaire.

**Enchaînement**

Le système affiche la page d’authentification.

L’utilisateur saisit son login et mot de passe et choisit le module prévue « horaire »

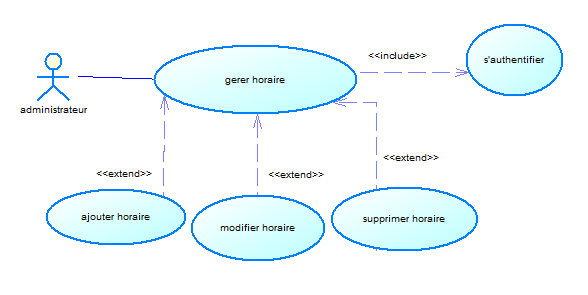
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’utilisateur fournit des paramètres d’identification inconnus, le système affiche un message d’erreur et demande de saisir un nouveau mot de passe.

La figure 9 présente le diagramme de cas d’utilisation « gérer horaire »



**Figure 9 : Diagramme de cas d'utilisation «gérer horaire»**

* **Gérer groupe**

**Pré condition**

L’administrateur doit s’authentifier pour qu’il puisse gérer les groupes.

**Enchaînement**

Le système affiche la page de l’authentification.

L’utilisateur entre son login et mot de passe et choisit le module prévue « groupe».

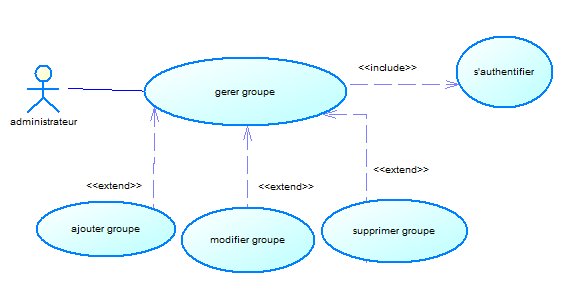
Le système valide les informations d’authentification par rapport à celles qui sont mémorisées.

Le système ramène l’utilisateur à la page demandée.

**Post condition**

Lorsque l’identification de l’utilisateur est inconnue, le système demande de saisir une nouvelle authentification.

La figure 12 sous-dessous apparaître le diagramme de cas d’utilisation associé au cas d’utilisation « gérer groupe »

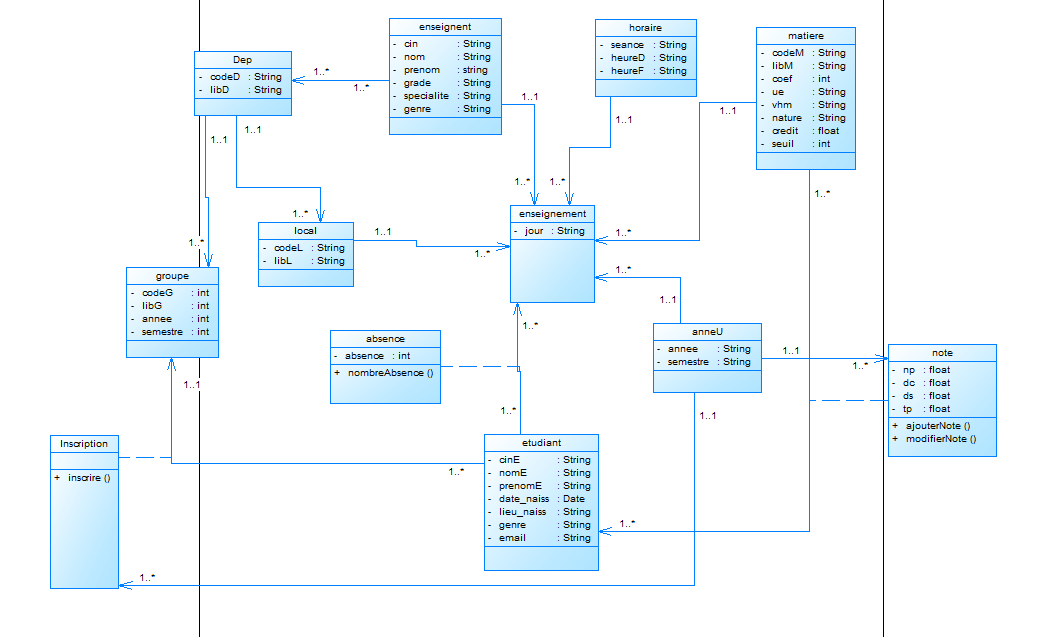


**Figure 12 : Diagramme de cas d'utilisation « gérer groupe»**

## Diagramme de classe

Cette phase va préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d’analyse. La technique utilisée pour identifier les classes candidates est la suivante :

* Chercher les noms communs importants dans les descriptions textuelles des cas d’utilisation.
* Vérifier les propriétés « objet » de chaque concept (identité, propriétés, comportement), puis définir ses responsabilités.



**Figure 13 : Diagramme de classe**

## Diagrammes de séquence

Les diagrammes de séquence d’UML sont des diagrammes qui représentent les échanges de messages entre objets, dans le cadre d’un fonctionnement particulier du système. Ils servent à développer en analyse les scénarios d’utilisation du système.

# Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la démarche conceptuelle que nous avons adoptée pour la réalisation de notre solution. Dans un premier temps nous avons exploré, dans une conception préliminaire, l’architecture globale de l’application pour passer à le détailler. Ensuite nous avons abordé de manière précise les relations entre les différents composant du système et leurs interactions et ce à l’aide des différents diagrammes de classes et des diagrammes de séquences.

Nous aborderons dans le chapitre suivant la partie réalisation où nous mettons en valeur le travail réalisé.

# 

*Chapitre III*

# Réalisation

## Introduction

Ce chapitre constitue le dernier volet de ce rapport, il traduit la phase qui a pour objectif l’implémentation de notre application. Nous commençons, tout d’abord, par la description de l’environnement matériel et logiciel utilisé pour développer cette application, ensuite nous présentons nos choix technologiques utilisés et enfin nous donnons un aperçu sur le travail réalisé.

# Environnement de développement

Nous présenterons dans ce qui suit les choix techniques qui nous ont aidés à élaborer notre solution, le matériel utilisé lors de la réalisation de l’application et les environnements logiciels avec lesquels nous avons assuré le développement.

## Environnement matériel

Le développement de l’application nécessite l’existence d’un environnement matériel spécifique qui est composé de :

• Ordinateur dont les caractéristiques sont comme suivies :

**Marque** : TOSHIBA

**Processeur** : Intel (R) Core (TM) i5-2467M CPU

**Mémoire** : 6 Go de RAM

**Disque Dur** : 120 Go SSD

**Système d’exploitation** : Windows 10

## Environnement logiciel

Le long de la phase de développement, nous avons utilisé l’environnement de développement suivant :

• Outils de développement : VSCODE

• MySQL : Serveur base de données MySQL

•Sybase (Power AMC) : Pour la modélisation des besoins et la conception en UML

• Microsoft Office 2010 : pour La rédaction du présent rapport a été réalisée avec Microsoft en utilisant l’éditeur Microsoft Word

## Choix technique

Nous exposons dans ce paragraphe les choix techniques que nous avons effectué pour la mise en œuvre de notre solution.

## Choix du langage HTML5

H TML5 est non seulement un langage de programmation puissant conçu pour être sûr, inter plate-forme et international, mais aussi un environnement de développement qui s’étend continuellement pour fournir des nouvelles caractéristiques et bibliothèques permettant de gérer de manière élégante des problèmes traditionnellement complexes dans les langages de programmation classiques, les accès aux bases de données, il composé par des langages comme le HTML, JavaScript, CSS3 ainsi que le langage PHP . Du point de vue du programmeur, HTML5 permet de réduire le temps de développement d’une application grâce à la réutilisation du code développé. Ce sont les raisons pour lesquelles nous avons choisi ce langage.

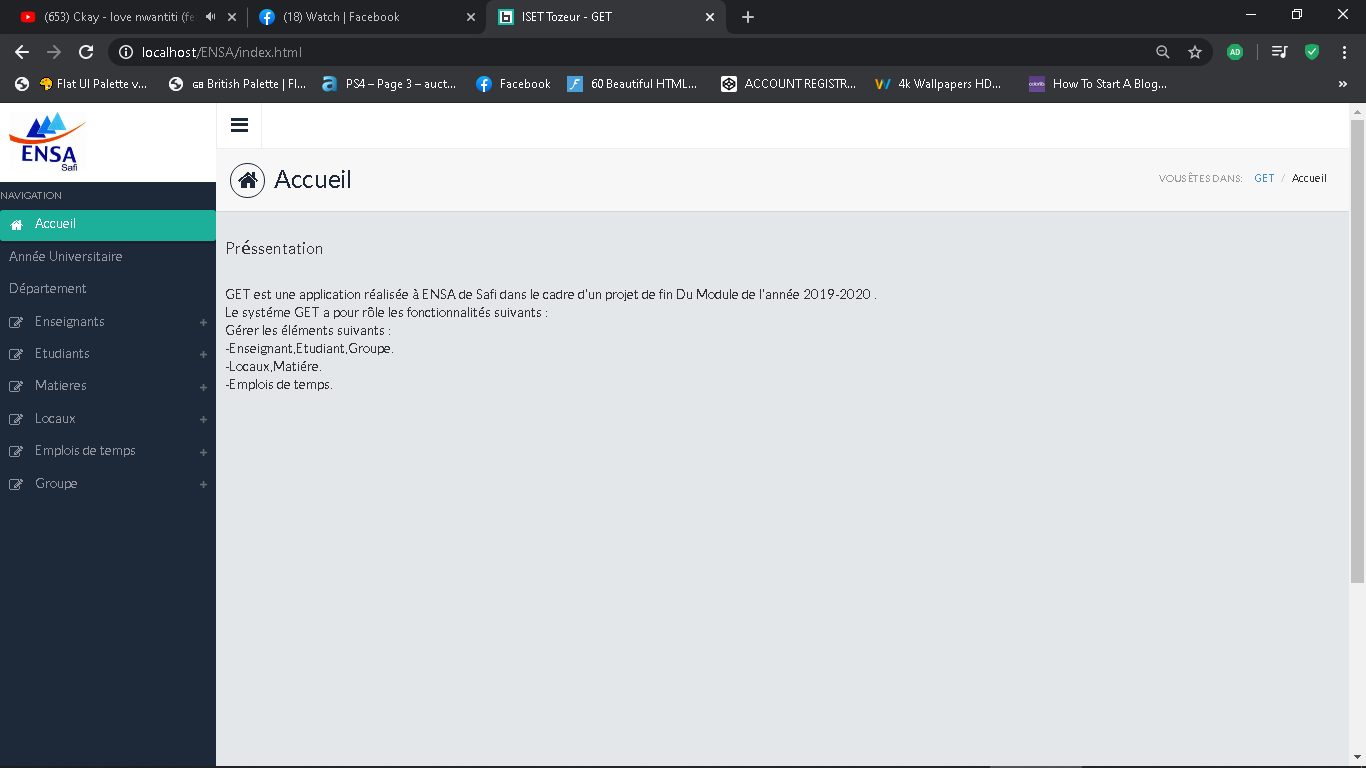
# Aperçu sur le travail réalisé

Les interfaces Homme/Machine constituent un élément dans la réussite d’une application.

Ainsi, nous essayerons dans cette partie de présenter les interfaces les plus significatives de notre application.

## Interface d’accueil

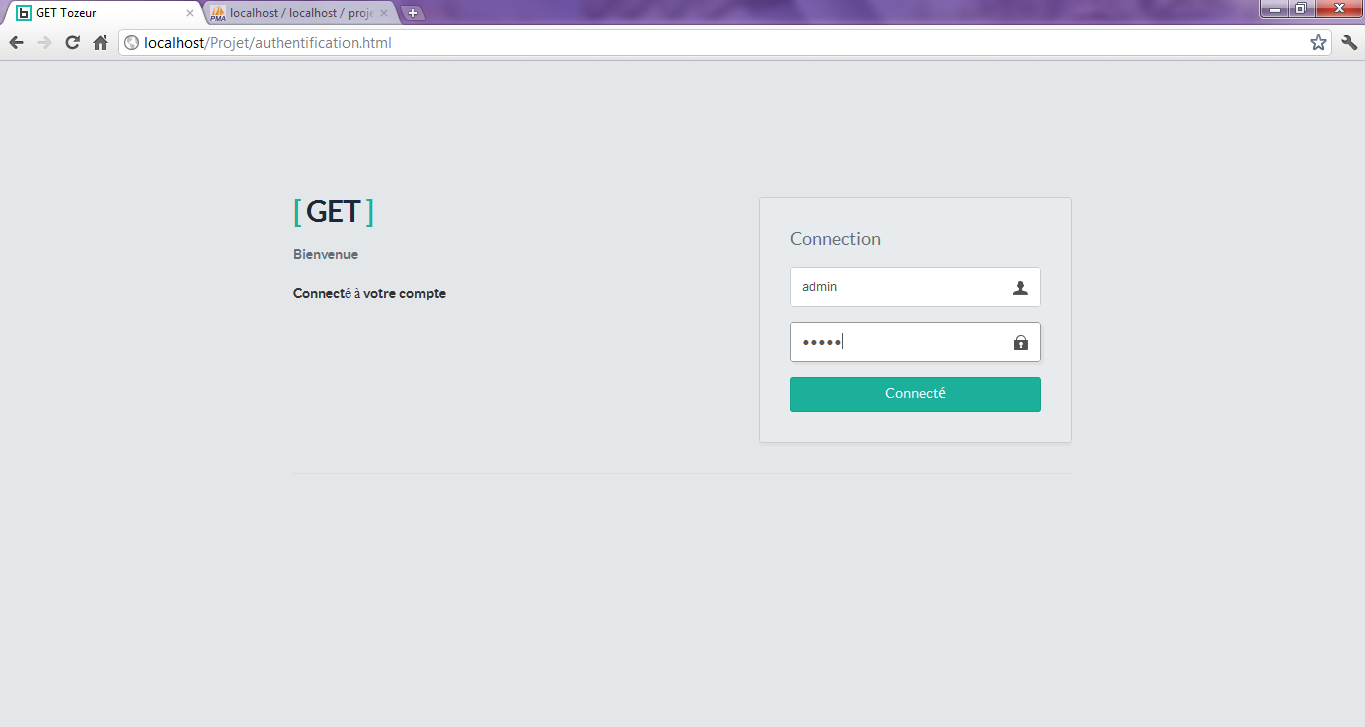
Le client est invité à lire une description de l’application dans la page d’accueil.



**Figure 16 : interface d’accueil**

## Interface d’authentification

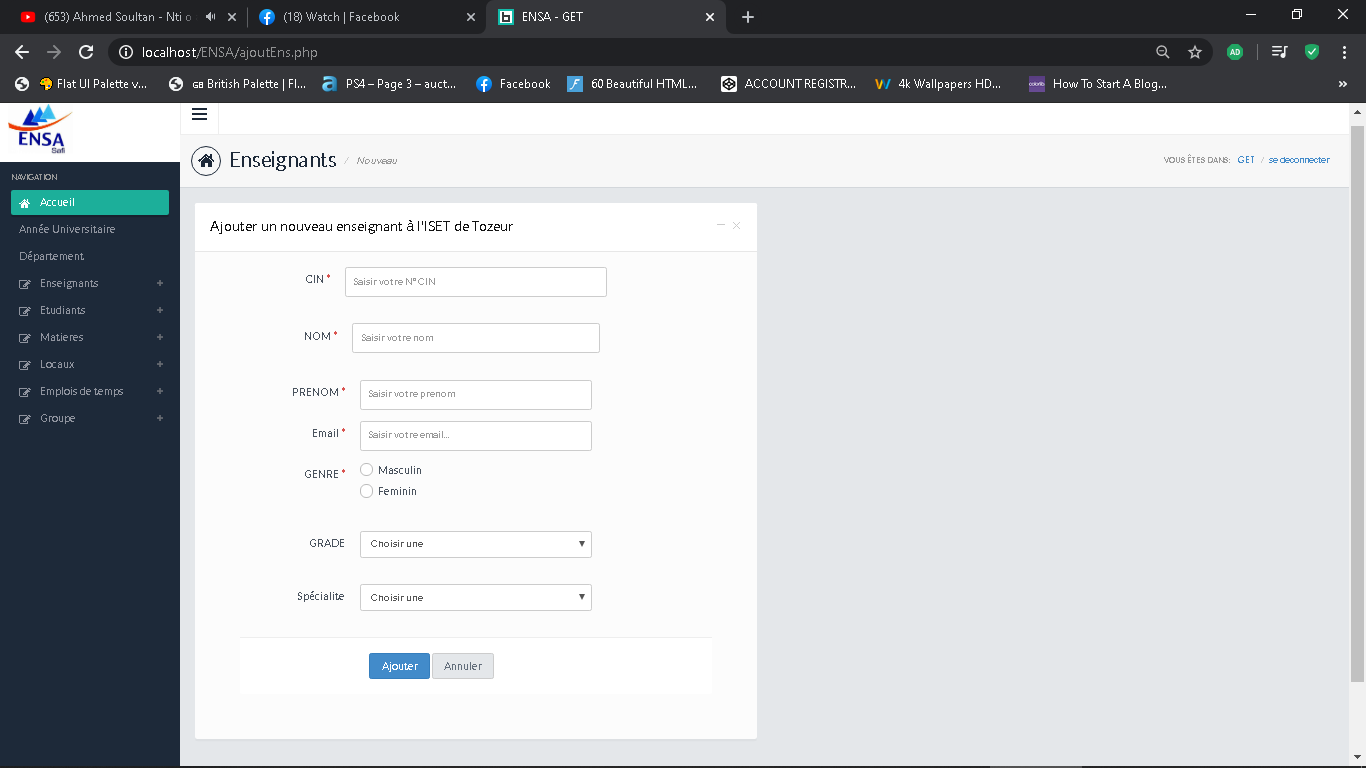
**Pour que l’utilisateur se connecte à son compte, il faut qu’il choisisse le menu « Connection » donc, une interface de connexion s’ouvre qui permette la connections.**



**Figure 17 : interface d’authentification**

## Interface d’ajouter un enseignant

Après que l’utilisateur a choisi le menu « enseignant» puis «Nouveau» dans la liste de menu, un formulaire sera affiché. Ce formulaire sera rempli par les informations personnelles des nouveaux enseignants, puis l’insérer dans la base de données.

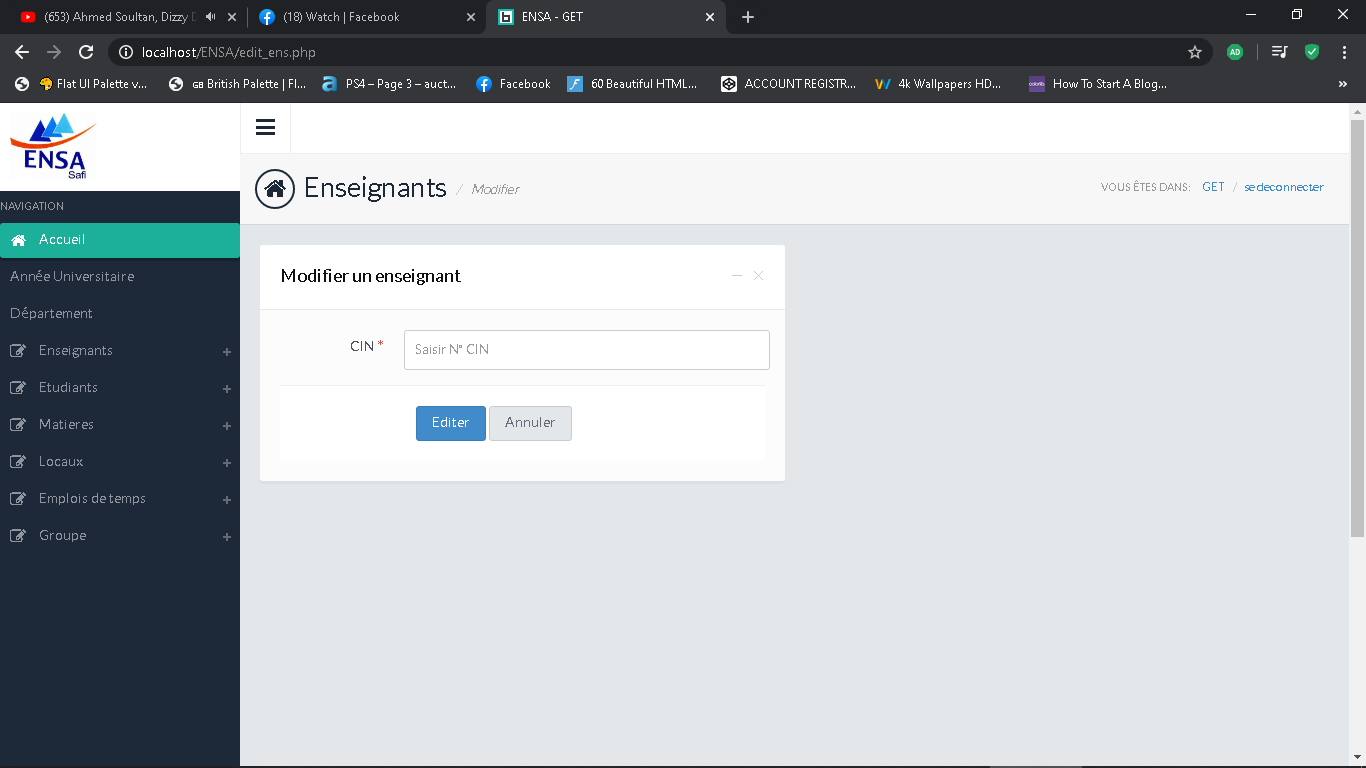


**Figure 18 : interface 1 d’ajouter enseignant**

**Après le remplissage de formulaire et le clique sur le bouton « Ajouter», une autre interface sera affiché qui indique si l’insertion a était effectué ou non et demande si l’utilisateur veut** insérer un nouveau enseignant.

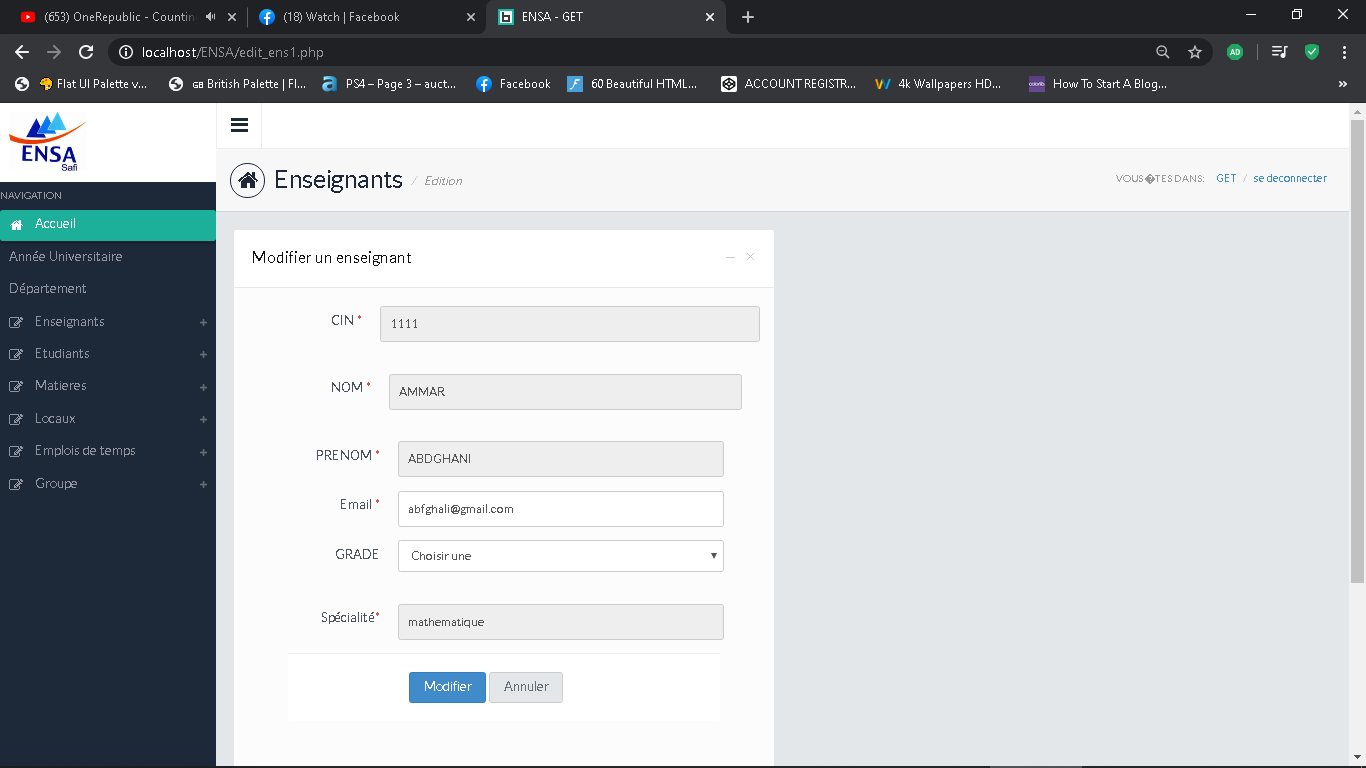
## Interface de modifier un enseignant

Après que l’utilisateur a choisi le menu « enseignant » puis « Edition» dans la liste de menu, un formulaire sera affiché .Ce formulaire sera rempli par le numéro CIN de l’enseignant à modifier.



**Figure 20 : interface 1 Edition enseignant**

Une autre interface sera affichée lorsqu’on clique sur le bouton « Editer » qui contient les informations qui seront modifié.



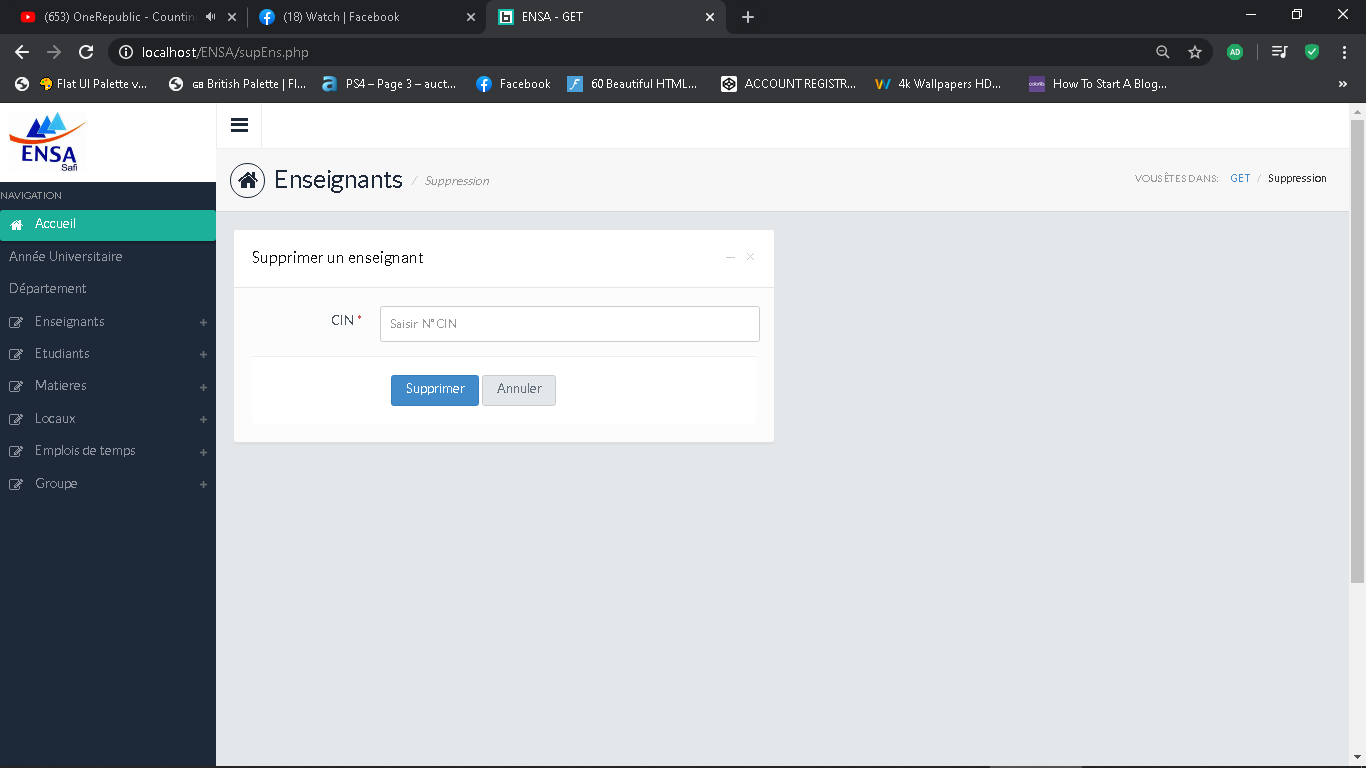
**Figure 21 : interface 2 Edition enseignant**

**Après la modification et le clique sur « modifier » une autre interface s’affiche qui indique si l’insertion a était effectuée ou non.**

**Figure 22 : interface 3 Edition enseignant**

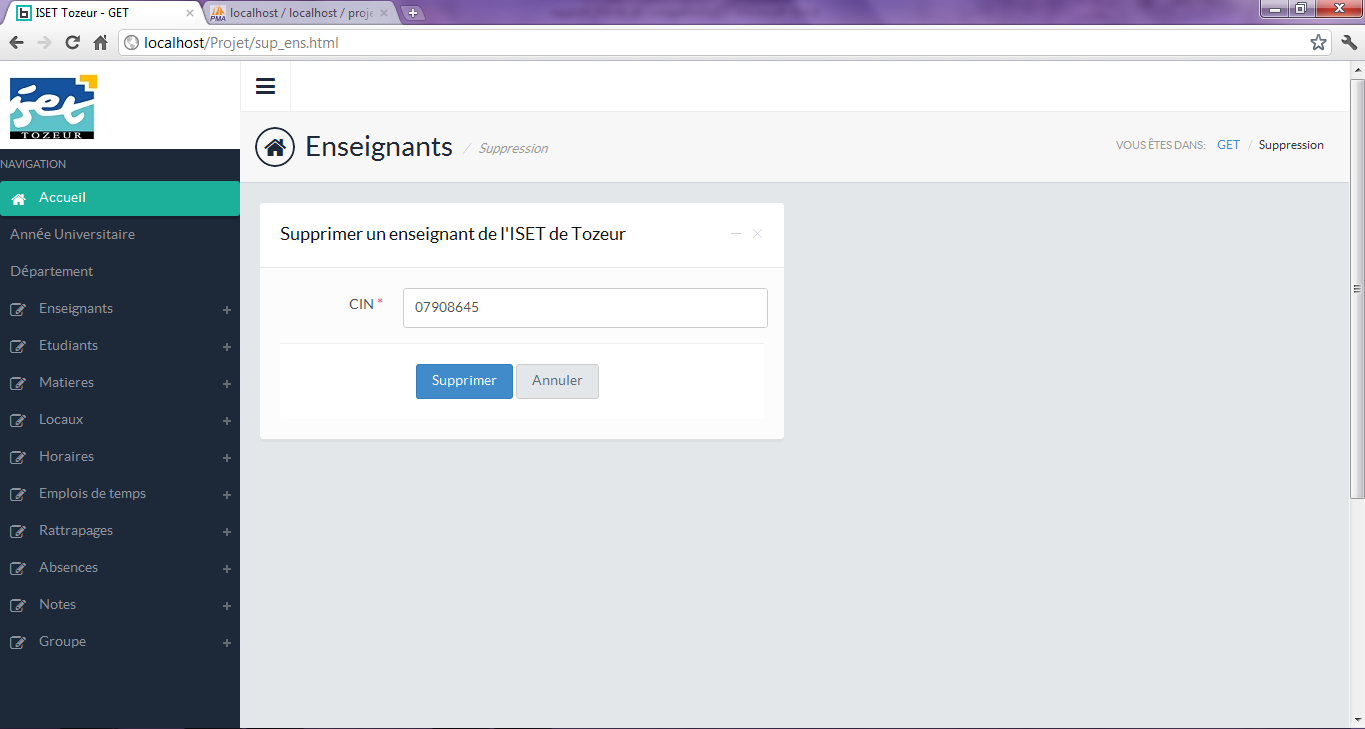
## Interface de supprimer un enseignant

Après que l’utilisateur a choisi le menu « enseignant » puis « Suppression » dans la liste de menu, un formulaire sera affiché. Ce formulaire sera rempli par le numéro CIN de l’enseignant a supprimé. Le saisi de numéro CIN a pour but de la sécurité.



**Figure 23 : interface 1 de Suppression enseignant**

Après la clique sur le bouton supprimer une interface sera affichée qui vous indiqué que la suppression a était effectué avec succée ou non effectué et vous demande aussi si vous voulez supprimer encore d’autre enseignant.



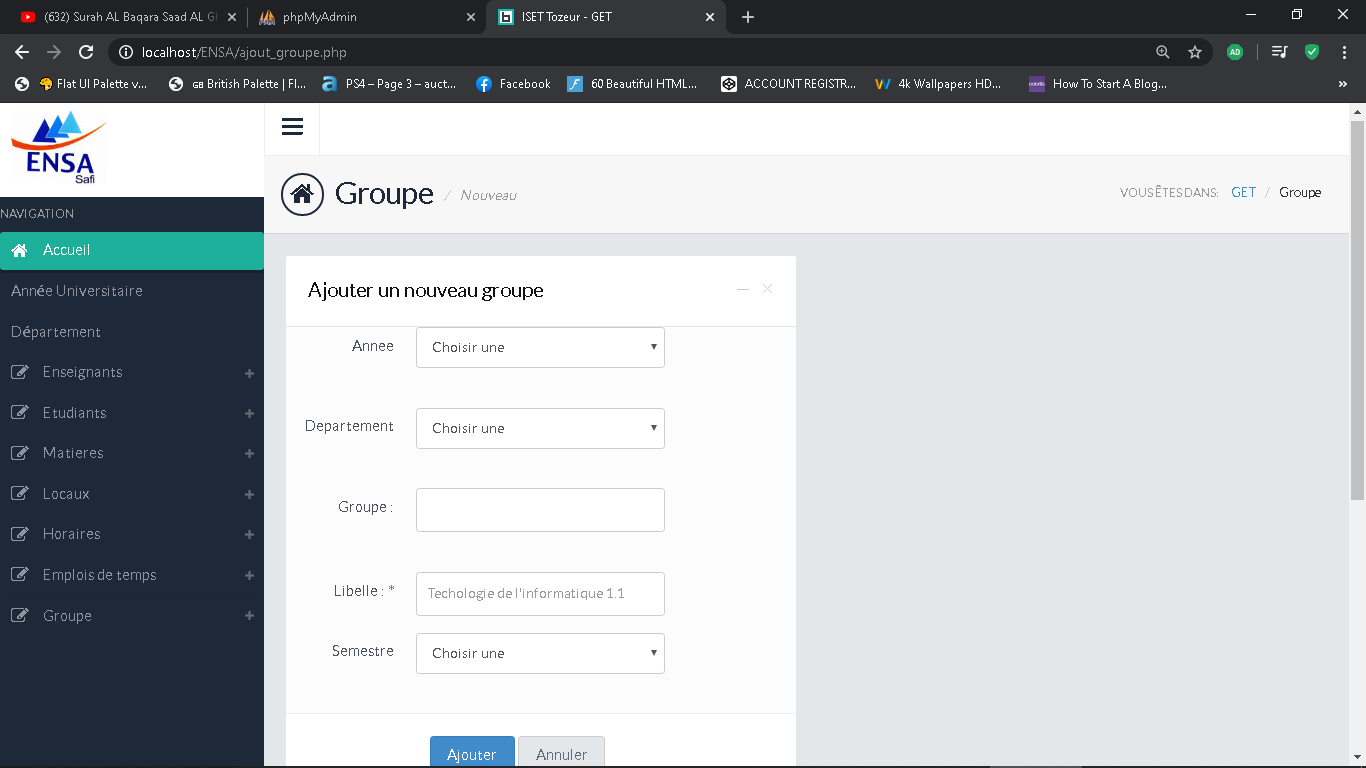
**Figure 24 : interface 2 de Suppression enseignant**

**N.B :** Les interfaces d’Etudiant, Matière et locaux ressemblent aux interfaces Enseignant.

## 

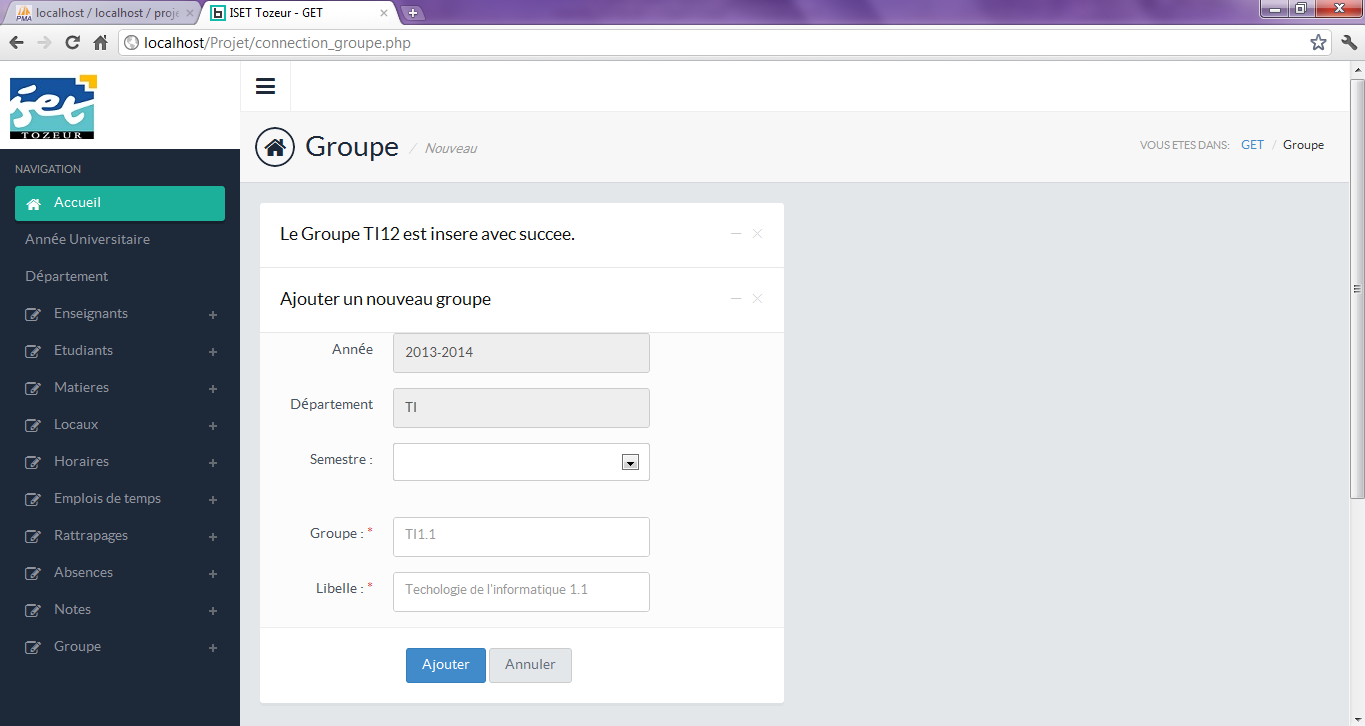
## Interface d’ajouter un groupe

Après que l’utilisateur a choisi le menu « groupe» puis «Nouveau» dans la liste de menu, un formulaire sera affiché .Ce formulaire sera rempli par les informations de nouveau groupe, puis l’insérer dans la base de données



**Figure 34 : Interface 1 Ajout groupe**

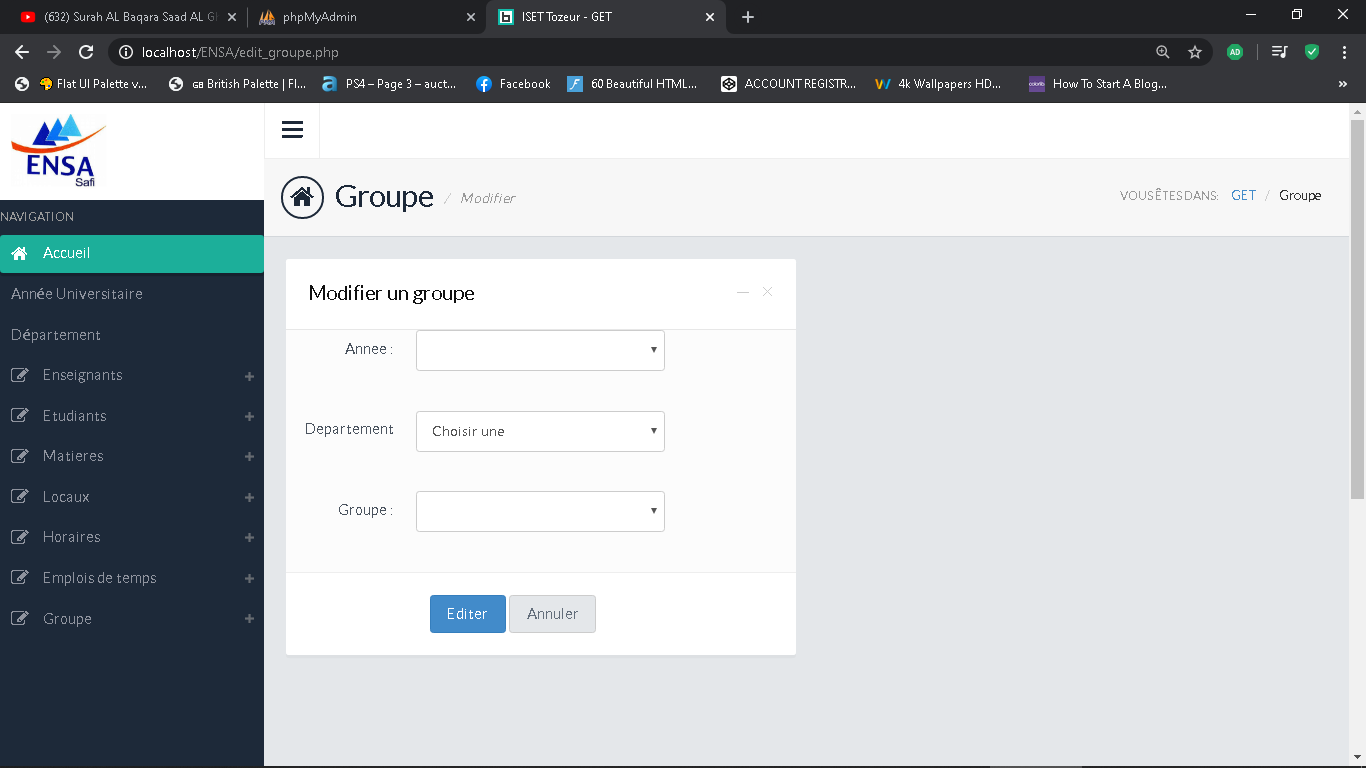
**Lorsque nous cliquons sur le bouton « Ajouter » une nouvelle interface sera affichée qui nous indique que l’insertion bien effectuée ou non et nous demande aussi si nous voulons ajouter d’autre groupe.**

****

**Figure 35 : Interface 2 Ajout groupe**

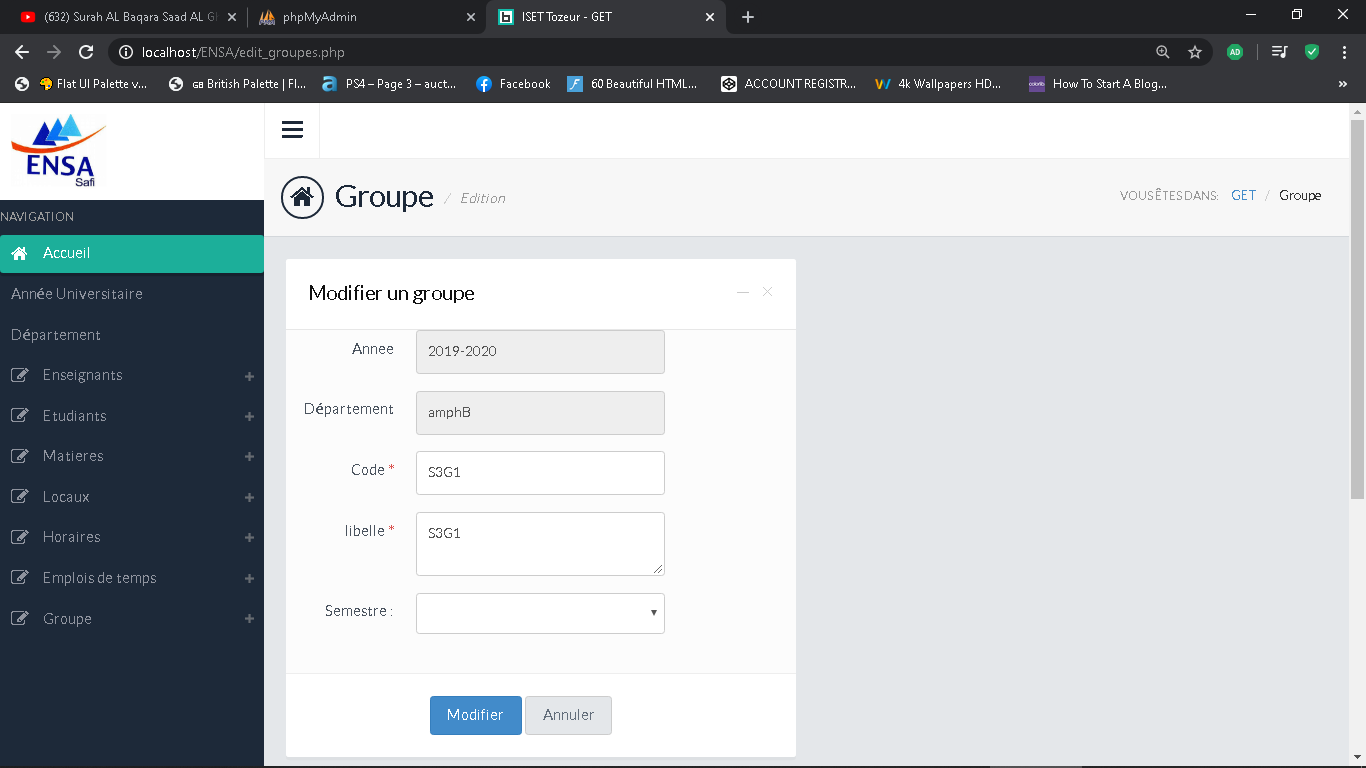
## Interface d’ajouter un groupe

Après que l’utilisateur a choisi le menu «groupe» puis «Edition» dans la liste de menu, un formulaire sera affiché .Ce formulaire sera rempli par le groupe a modifié



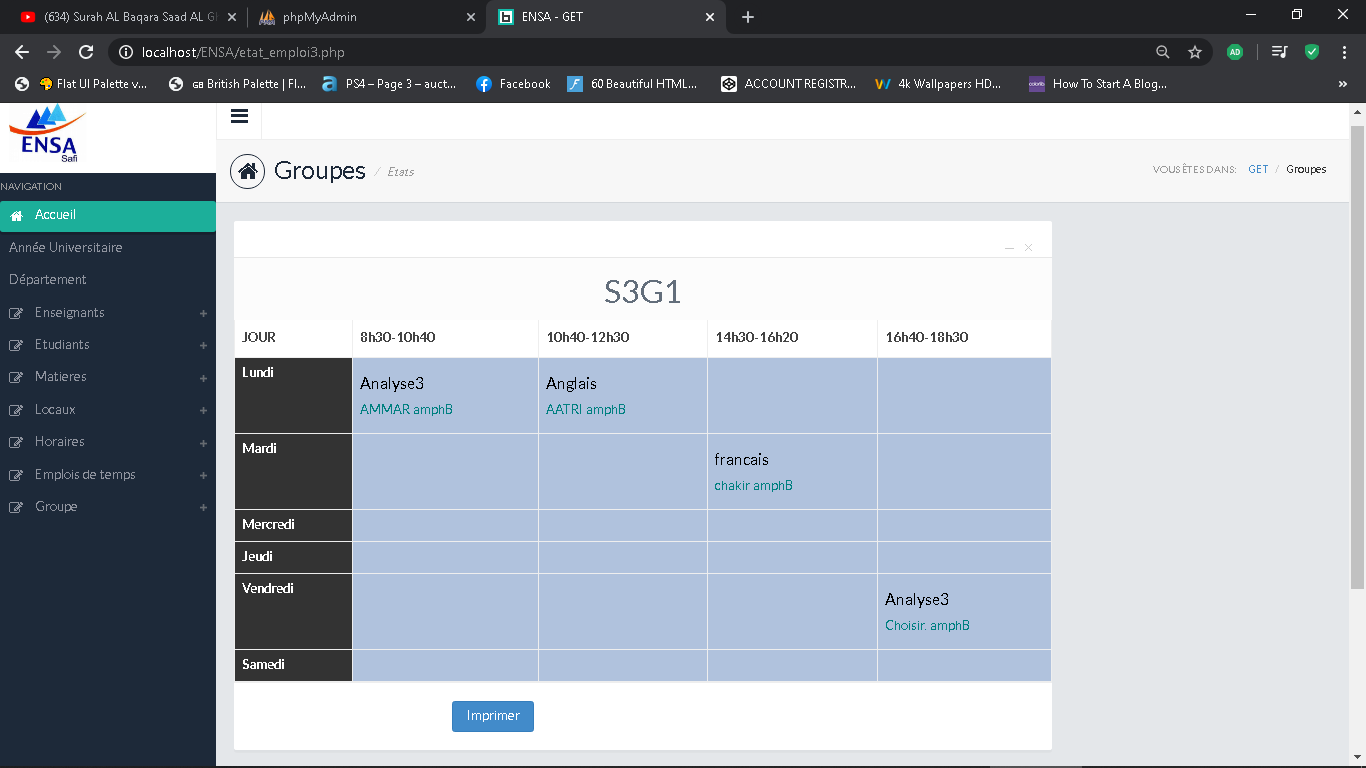
**Figure 36 : Interface 1 modifier groupe**

**Lorsque nous cliquons sur le bouton «Editer» une  autre interface sera affichée dans laquelle les informations qui seront modifier.**



**Figure 37 : Interface 2 modifier groupe**

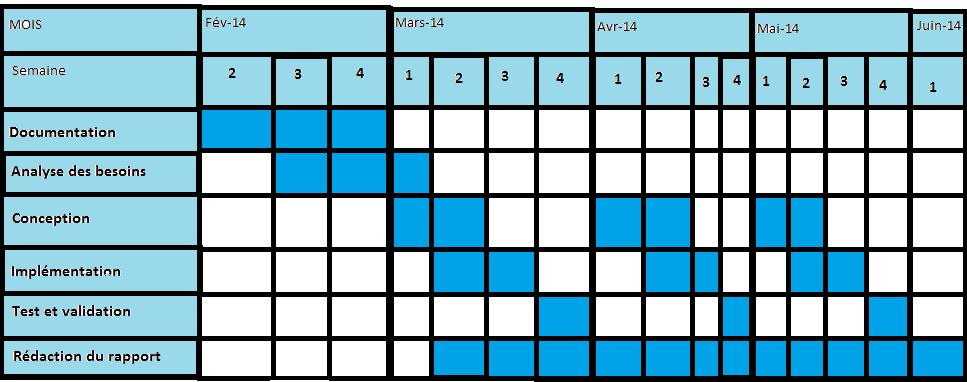
**.**



**Figure 38 : Interface d’afficher emplois**

## Chronogramme

Ce projet a été réalisé du 6 Février jusqu’au 31 Mai 2014. La figure sous-dessous représente le chronogramme de notre application décrivant la répartition des tâches du projet tout au long de ces quatre mois de stage.



**Figure 39 : Le chronogramme du travail**

## Conclusion

Au cours de ce chapitre, nous avons décrit les plateformes matérielles et logicielles sur lesquelles nous avons construit notre application. Nous avons, ensuite, présenté les interfaces les plus significatives de notre application. Enfin nous avons clôturé ce chapitre par la présentation du chronogramme d’activités.

# Conclusion Générale

Le domaine des développements connait depuis des années une explosion dans les fonctionnalités matérielles et les services offerts. En mois d’une décennie, les applications ont envahi le monde des universités et sont devenus indispensable dans le management de l’université et des ressources en permettant en même temps la performance, la mobilité et la simplicité.

# Bibliographie & Néographie

[**B1]** Modélisation objet avec UML : auteurs Pierre-Alain MLLEER & Nathalie GAERTNER

[**B2]** UML Modélisation des objets : auteurs Laurent DEBRAUWER & Fien VAN HEYDE

**[N1]** <http://www.edt-soft.com/>

**[N2]** <http://fr.openclassrooms.com/mooc/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

**[N3]** <http://www.memoireonline.com/sommaires/stage.html>

**[N4]** <http://baudet.me/2013/05/installer-pear-sur-son-wamp/>