

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Université Assane Seck de Ziguinchor
UFR Sciences et Technologies
Département Informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de Master
Mention : Informatique ; Spécialité : Génie Logiciel

Sujet :

ETUDE ET MISE EN PLACE D'UNE APPLICATION
DE GESTION ET SUIVI DES COURS A L'UASZ
« Le Cahier de Texte Electronique »

Présenté par : M. Moussa TOURE

Soutenance le 19/02/21

Sous la direction de : Dr Thierno Ahmadou DIALLO

Sous la supervision du : Pr. Youssou FAYE

Membres du jury

Youssou FAYE	Professeur Assimilé	Président	UASZ
Thierno Ahmadou DIALLO	Maître de Conférences Assimilé	Encadreur	UASZ
Ibrahima DIOP	Maître de Conférences	Rapporteur	UASZ
Youssou Dieng	Maître de Conférences	Rapporteur	UASZ

Année académique 2019-2020

Résumé

La gestion et le suivi des cours à l'Université Assane Seck de Ziguinchor ont toujours été un sujet de réflexion. Pratiquement, chaque année, le personnel administratif et personnel enseignant et de recherche s'engagent à mener les cours dans les meilleures conditions. Pour bien assumer la tâche, chaque classe dispose d'un registre appelé cahier de texte. Malgré ces efforts, le problème de la gestion et du suivi des cours persiste toujours. Ainsi pour alléger la tâche aux acteurs concernés, nous souhaitons automatiser la gestion et le suivi des cours. Autrement dit, mettre en place un cahier de texte électronique.

Dédicaces

Nous dédions ce travail :

- ♣ A nos très chers parents, pour tous les sacrifices consentis ;
- ♣ Aux personnes qui nous ont soutenus durant tout notre cursus scolaire et universitaire ;
- ♣ A nos frères et sœurs qui nous ont tendu la main et sans cesse encouragés ;
- ♣ A tous nos amis qui nous ont assistés et soutenus.

Remerciements

Après avoir rendu grâce à ALLAH, Le Tout Puissant, Le Tout Miséricordieux de nous avoir donné l'opportunité de finir ce travail colossal qui nous a été confié, nous voulons particulièrement, à travers cette liste non exhaustive de personnes et d'enseignants chercheurs, remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation du présent document ; nous voulons nommer :

Les membres du jury qui ont accepté d'évaluer ce travail :

- ♣ Pr Youssou FAYE comme président du jury.
- ♣ Dr Youssou DIENG et Dr Ibrahima DIOP comme rapporteurs.

Nous révélons notre gratitude à notre encadreur, le Dr Thierno Ahmadou DIALLO qui a été très disponible durant toute la durée du mémoire. Nous vous remercions également pour avoir accepté de nous encadrer, vous avez fortement contribué à la conception et à la réalisation du présent document. Soyez assuré de notre entière satisfaction.

Nos sentiments de reconnaissance s'adressent également au corps enseignant de l'Université Assane Seck de Ziguinchor ainsi qu'à tous ceux qui ont contribué moralement et financement à notre formation durant les cinq années d'études.

Enfin, nous ne pouvons pas clore cette partie sans remercier nos amis, camarades et tous ceux qui, de près ou de loin, ont contribué à l'accomplissement de cette œuvre.

Table des matières

RESUME	i
DEDICACES	ii
REMERCIEMENTS	iii
LISTE DES FIGURES	vi
LISTE DES TABLEAUX	viii
LISTE DES ABREVIATIONS	ix
INTRODUCTION GENERALE	1
CHAPITRE 1 CONTEXTE JUSTIFICATIF DU SUJET	2
1.1. PRESENTATION DE L'UASZ.....	2
1.2. PRESENTATION DU CAHIER DE TEXTE.....	3
1.3. PROBLEMATIQUE DU SUJET	3
1.3.1. Solution proposée	4
1.3.2. Objectif attendu	4
1.4. CONCLUSION	4
CHAPITRE 2 PROCESSUS DE DEVELOPPEMENT DE L'APPLICATION	5
2.1. LA METHODOLOGIE SCRUM.....	5
2.1.1. Les différents rôles de la méthodologie Scrum	7
2.1.2. Justification du choix sur Scrum	8
2.1.3. Organisation de l'équipe.....	8
2.2. CONCLUSION	8
CHAPITRE 3 SPECIFICATION ET L'ANALYSE DES BESOINS FONCTIONNELS	9
3.1. SPECIFICATION DES BESOINS FONCTIONNELS.....	9
3.1.1. Identification des acteurs du système	9
3.1.2. Identification des fonctionnalités du système.....	10
3.1.3. Diagramme de cas d'utilisation	12
3.2. ANALYSE DES BESOINS FONCTIONNELS DU SYSTEME	17
3.2.1. Analyse de l'authentification.....	17
3.2.2. Analyse de la gestion administrative du système	20
3.3. CONCLUSION	22
CHAPITRE 4 CONCEPTION DU SYSTEME	23
4.1. CONCEPTION GENERALE.....	23
4.1.1. Architecture de l'application	23
4.1.2. Diagramme de composant	24
4.1.3. Diagramme de package	24
4.1.4. Diagramme de déploiement.....	25
4.2. CONCEPTION DETAILLEE	25

4.2.1.	Dictionnaire des données.....	25
4.2.2.	Diagramme de classe	25
4.2.3.	Diagramme de classe participante à la gestion de l'authentification des utilisateurs	26
4.2.4.	Diagramme de classe participante à la gestion des cours	27
4.3.	CONCLUSION	27
CHAPITRE 5 IMPLEMENTATION ET PRESENTATION DE L'APPLICATION		28
5.1.	TECHNOLOGIES UTILISEES.....	28
5.1.1.	Xampp	28
5.1.2.	Power AMC.....	28
5.1.3.	Uml	28
5.1.4.	JavaScript	29
5.1.5.	Bootstrap.....	29
5.1.6.	HTML5 / CSS3.....	30
5.1.7.	PHP.....	30
5.1.8.	Framework CodeIgniter.....	30
5.1.9.	TCPDF.....	31
5.1.10.	PHPExcel.....	31
5.1.11.	Outil de développement de l'application : Netbeans	31
5.2.	IMPLEMENTATION.....	32
5.2.1.	Installation et configuration du Framework CodeIgniter	32
5.2.2.	Implémentation de la base de données	33
5.2.3.	Implémentation de l'application.....	36
5.2.4.	Présentation de quelques interfaces de l'application	38
5.3.	CONCLUSION	50
CONCLUSION GENERALE ET PERSPECTIVES.....		51
ANNEXES		i
ANNEXE A : DICTIONNAIRE DE DONNEES		i
ANNEXE B : CAHIER DE CHARGE.....		iii
BIBLIOGRAPHIE		v
WEBOGRAPHIE		v

Liste des figures

Figure 1 : Fonctionnement de Scrum	7
Figure 2 : Exemple de représentation d'un acteur.....	9
Figure 3 : Diagramme de cas d'utilisation de l'administrateur	13
Figure 4 : Diagramme de cas d'utilisation des responsables pédagogiques	14
Figure 5 : Diagramme de cas d'utilisation des responsables de formation et chefs de département....	15
Figure 6 : Diagramme de cas d'utilisation de l'étudiant.....	16
Figure 7 : Diagramme de cas d'utilisation de l'enseignant	16
Figure 8 : Diagramme de séquence du cas d'utilisation « s'authentifier ».....	18
Figure 9 : Diagramme d'activité du cas d'utilisation « s'authentifier »	19
Figure 10 : Diagramme de séquence du cas d'utilisation « ajout utilisateur »	21
Figure 11 : Diagramme d'activité du cas d'utilisation « ajout utilisateur »	22
Figure 12 : Architecture de l'application	23
Figure 13 : Diagramme de composant.....	24
Figure 14 : Diagramme de package	25
Figure 15 : Diagramme de déploiement	25
Figure 16 : Diagramme de classe	26
Figure 17 : Diagramme de classe participante à l'authentification des utilisateurs	27
Figure 18 : Diagramme de classe participante à la gestion des cours.....	27
Figure 19 : Page d'accueil après l'installation de CodeIgniter.....	32
Figure 20 : Capture d'écran du contenu du répertoire de notre application	32
Figure 21 : Modèle physique des données.....	34
Figure 22 : Capture d'écran après la création de la base de données.	34
Figure 23 : Importation de la base de données.	35
Figure 24 : Capture d'écran des tables de la base de données.....	36
Figure 25 : Le fichier de configuration pour la connexion vers la base de données	36
Figure 26 : La classe « enseignant » : Modèle	37
Figure 27 : La classe « enseignant » : Vue	37
Figure 28 : La classe « enseignant » : Contrôleur	38
Figure 29 : La page d'authentification	39
Figure 30 : La page d'accueil de l'administrateur.....	40
Figure 31 : La page d'accueil du responsable de formation	41

Figure 32 : La page gestion enseignant et matière du responsable de formation	41
Figure 33 : La page d'accueil du responsable pédagogique.....	42
Figure 34 : La page d'accueil de l'étudiant.....	43
Figure 35 : La page gestion enseignant et cours pour l'étudiant	43
Figure 36 : La page gestion enseignant et cours pour le responsable de la classe.....	44
Figure 37 : La page gestion cours et évaluation pour le responsable de la classe	45
Figure 38 : La page d'accueil enseignant.....	46
Figure 39 : La page impression fiche de déclaration.....	46
Figure 40 : La page validation.....	47
Figure 41 : La page gestion cours et évaluation	48
Figure 42 : La page impression fiche de déclaration détaillée	48
Figure 43 : La page étudiant.....	49
Figure 44 : Capture d'écran du contenu du fichier Excel.....	49
Figure 45 : La page envoyer mail.....	50
Figure 46 : Tableau de bord.....	iv

Liste des tableaux

Tableau 1: Exemple d'un « user story ».....	5
Tableau 2 : Présentation de l'équipe projet.....	8
Tableau 3 : Identification des acteurs	9
Tableau 4 : Identification des fonctionnalités.....	10
Tableau 5 : Description du cas d'utilisation « s'authentifier ».....	17
Tableau 6 : Description du cas d'utilisation « gérer utilisateur ».....	20

Liste des abréviations

UASZ :	Université Assane Seck de Ziguinchor
CM :	Cours Magistral
TP :	Travail Dirigé
TD :	Travail Pratique
UFR :	Unité de Formation et de Recherche
LMD :	Licence Master Doctorat
UML :	Unified Modeling Language
PATS :	Personnel Administratif, Technique et de Service
PER :	Personnel Enseignant et de Recherche
PATSF :	Personnel Administratif, Technique et de Service Fonctionnaire
CTEXTE :	Cahier de texte électronique
ANSSI :	Agence nationale de la Sécurité des Systèmes d'Information
FOAD :	Service de Formation Ouverte et à Distance (FOAD)
CRI :	Centre des Ressources Informatiques (CRI)
BU :	Bibliothèque Universitaire (BU)
DES :	Direction du service aux Etudiant (DSE)
DAF :	Direction des Affaires Financières (DAF)
DCS :	Direction Centrale de la Scolarité (DCS)
DGPM :	Direction de la Gestion du Patrimoine et de Maintenance (DGPM)
DES :	Direction de l'Environnement et de la Sécurité (DES)
DRH :	Direction des Ressources Humaines (DRH)

Introduction générale

L'informatique, science de traitement automatique de l'information, constitue un domaine pratiquement incontournable pour la résolution de multiples problèmes, principalement ceux liés à la gestion optimale des organisations. Par exemple, la répartition des enseignements à l'UASZ est un processus obligatoire qui s'effectue de façon périodique [23]. Selon le mode de fonctionnement, pour chaque semestre donné, tout enseignant doit connaître ses enseignements et le nombre d'heures total d'enseignement.

Cependant, pour une bonne gestion et suivi des cours, l'UASZ met à la disposition de chaque classe un registre appelé cahier de texte. En principe, à la fin du cours, le responsable de la classe ou l'enseignant est appelé à remplir le cahier de texte. Ce dernier est un document officiel à valeur juridique ainsi qu'un outil pédagogique important qui permet d'informer les étudiants, le personnel administratif ainsi que le personnel enseignant et qui atteste du travail fait en classe. Il faut noter également que cet outil pédagogique permet aussi de mesurer la charge de travail d'une classe et d'inciter le personnel administratif à mieux répartir dans le temps les demandes et les contrôles. Malgré son utilité, ce document présente d'énormes irrégularités vis-à-vis de sa gestion en version papier.

Pour pallier ce manquement, nous abordons dans ce présent travail l'étude et la mise en place d'une **application de gestion et suivi des cours**. Autrement dit, la mise en place d'un cahier de texte électronique.

Le projet **CTEXTE** a pour objectif d'automatiser le cahier de texte de la classe afin de permettre aux étudiants, de pouvoir y accéder, de consulter les cours et leurs états d'avancement, mais aussi de mettre à leur disposition les supports de cours sans difficulté. Cette application permettra également aux personnels administratifs (responsable de la formation, Chef de département, responsable pédagogique) et aux personnels enseignants (permanent et vacataire) de travailler dans de meilleures conditions.

Pour présenter le travail réalisé, nous allons d'abord décrire le contexte justificatif du sujet avant d'expliquer le processus de développement de l'application. Ensuite nous présenterons la spécification et l'analyse des besoins fonctionnels. Puis, nous aborderons la partie conception du système. Enfin, nous décrirons, en dernière partie, l'implémentation et la présentation de l'application.

Chapitre 1 Contexte justificatif du sujet

1.1. Présentation de l'UASZ

L'UASZ est créée par le décret 2008-537 du 22 mai 2008 comme troisième Université du Sénégal [18]. Elle fonctionne sous le système Licence Master Doctorat (LMD) et offre de nombreuses formations. Elle a démarré ses activités en février 2007 avec trois UFR (Unité de Formation et de Recherche), que sont :

- ♣ l'UFR des Sciences Economiques et Sociales ;
- ♣ l'UFR des Sciences et technologies ;
- ♣ l'UFR des Lettres, Arts et Sciences Humaines.

En 2011, une nouvelle UFR a été ajoutée : il s'agit de l'UFR des Sciences de la Santé. Chacune des UFR dispose d'un service chargé de la pédagogie géré par un chef de service pédagogique, et de départements qui sont gérés par des chefs de département. Ces départements proposent des formations qui sont gérées par des responsables de formation. Pour la bonne gestion de ces quatre entités (UFR, services pédagogiques, départements et formations) l'UASZ dispose :

- ♣ d'un Personnel Administratif, Technique et de Service (PATS) ;
- ♣ d'un Personnel Enseignant et de Recherche (PER).

De nombreux centres et services communs sont notés et ils ne cessent d'augmenter d'année en année. Ainsi, on distingue :

- ♣ le Service de Formation Ouverte et à Distance (FOAD);
- ♣ le Centre des Ressources Informatiques (CRI) ;
- ♣ la Bibliothèque Universitaire (BU);
- ♣ la Direction du service aux Etudiant (DSE);
- ♣ la Direction des Affaires Financières (DAF);
- ♣ la Direction Centrale de la Scolarité (DCS);
- ♣ la Direction de la Gestion du Patrimoine et de Maintenance (DGPM);
- ♣ la Direction de l'Environnement et de la Sécurité (DES);
- ♣ la Direction des Ressources Humaines (DRH).

Ainsi, en terme de pédagogie d'importante décision ont été prise au sein de l'UASZ. Par exemple, avant le début de chaque semestre, le processus de répartition des enseignements est enclenché. Il permet aux enseignants de connaître leur enseignement ainsi que le nombre

d'heure total pour chaque enseignement. Comme la réparation des enseignements nécessite un contrôle rigoureux. Dès lors, chaque année, l'université débloque une somme importante pour la confection de registre appelé cahier de texte. Pour plus de transparence sur le déroulement des cours, le cahier de texte est placé par la suite dans chaque classe et sous la tutelle du responsable de la classe. À la fin du cours, deux individus sont destinés à agir sur le cahier de texte : l'enseignant et le responsable de la classe.

1.2. Présentation du cahier de texte

En réalité, le cahier de texte est un document qui a toujours existé dans le système éducatif, mais la version papier demeure néanmoins un outil très ancien. C'est à travers ce document que le service administratif se base pour évaluer les cours dispensés par l'enseignant pour un semestre donné.

Toutefois, il arrive que, dans sa forme papier, cet outil subisse certains aléas :

- ♣ **Pertes ou dégradations** lors de manipulations peu délicates ;
- ♣ **Accessibilité réduite et consultation peu fréquente par les étudiants** : le cahier de texte n'est pas à la portée de tout le monde. En présence du cours, il est placé sous la responsabilité du responsable de la classe et en dehors de celui-ci, il est déposé au niveau de l'UFR.
- ♣ **Procédure de rémunération des enseignants complexes** : l'enseignant à la fin de chaque semestre, doit impérativement rédiger un rapport justifiant le travail fait. Ensuite, ce document est déposé auprès du chef de département concerné pour d'éventuelles vérifications.
- ♣ **Dérèglement du calendrier de gestion et suivi des cours, en cas de perturbation** : depuis des années, nous avons constaté que l'année universitaire demeure néanmoins instable. Du jour au lendemain, étudiant comme enseignant, décrète des grèves qui marquent l'arrêt total des cours provoquant ainsi une répercussion sur le calendrier de gestion et suivi des cours.

1.3. Problématique du sujet

Le système d'information actuel de l'UASZ et plus particulièrement le cahier de texte est non encore automatisé, toutefois la gestion et le suivi des cours se fait toujours de façon manuelle, ce qui rend la tâche fastidieuse et complexe aux acteurs concernés. Ces manquements énumérés ci-dessus ont de grandes répercussions sur la qualité du travail et par conséquent, un effet négatif sur les résultats attendus dans la gestion et le suivi des cours.

1.3.1. Solution proposée

Après avoir analysé le contexte du système actuel, nous avons jugé nécessaires de mettre en place une application web dotée d'une architecture 3-tiers pouvant remédier aux différentes lacunes soulevées dans la problématique. Elle permettra entre autres :

- ♣ **d'attribuer des comptes aux utilisateurs** : faire de tel sorte que chaque utilisateur du système que cela soit le chef de département, le responsable de formation, le responsable pédagogique, l'enseignant ou l'étudiant de disposer d'un compte et d'accéder à son espace de travail.
- ♣ **de contrôler le cahier de texte** : permettre aux personnels administratifs et personnels enseignants d'avoir la main mise sur le cahier de texte afin d'avoir une vue globale sur la chose.
- ♣ **de simplifier la procédure de rémunération aux enseignants** : ceci étant dit que l'enseignant au lieu de rédiger un rapport pour justifier le travail fait, il le générer de façon automatique.

1.3.2. Objectif attendu

Les principaux objectifs attendus dans l'automatisation de la gestion et du suivi des cours sont :

- ♣ d'attribuer des matières aux enseignants ;
- ♣ de superviser les cours et leur état d'avancement ;
- ♣ de consulter cours ;
- ♣ de télécharger les supports de cours ;
- ♣ de générer des fiches de déclaration des heures de cours justifiant le travail;
- ♣ de réajuster le calendrier de gestion et suivi des cours en cas de perturbation.

À la fin du projet, nous souhaitons obtenir une application web fiable, compréhensible et facile à utiliser pour les utilisateurs du système.

1.4. Conclusion

Cette première phase nous a permis d'avoir une vision plus claire de l'existant et des problèmes auxquels les différents acteurs sont confrontés. Ensuite, nous avons eu à poser des objectifs spécifiques pour résoudre les contraintes liées à la gestion du cahier de texte. Cependant, pour atteindre ces objectifs, nous avons besoin aussi d'une méthodologie de développement. Ainsi, dans la deuxième phase de notre travail, nous décrivons le processus de développement à suivre dans ce projet.

Chapitre 2 Processus de développement de l'application

Un processus de développement est un ensemble d'activités ou d'actions structurées dans le but d'atteindre l'objectif d'un projet. Dans le cadre d'un projet informatique, de nombreux facteurs interviennent de la planification à la gestion d'un projet comme le délai, le coût, la qualité, les ressources et la capacité organisationnelle. Une défaillance aperçue sur un des facteurs cités ci-dessous entraîne immédiatement des problèmes dans le déroulement du projet. Ces risques, figurant dans un projet, induisent les développeurs à utiliser un cadre de gestion de projet. Dans notre cas, nous allons utiliser la méthodologie Scrum qui est une des méthodes agiles.

2.1. La méthodologie Scrum

Scrum est un cadre de développement dans lequel des équipes plurifonctionnelles réalisent des produits de manière itérative et incrémentale [4]. Un projet Scrum est un projet où le client n'a pas encore défini toutes les fonctionnalités dont il a besoin. Ce qui permet donc une certaine souplesse d'organisation.

Le projet est organisé autour de « **sprints** » de développement (les itérations) d'une durée allant généralement de deux à quatre semaines. Le projet débute par le « **sprint 0** », dédié à la réalisation de tous les travaux de préparation et de mise en place : conception et architecture, environnements de développement, outils de suivi et d'intégration.

Les fonctionnalités demandées sont listées et décrites sous la forme de « **user stories** » et placées dans le « **backlog** » (carnet) de produit. Le contenu d'un « **user story** » ressemble à la description ci-dessous :

Tableau 1: Exemple d'un « user story »

ID :	A01
Scenario :	En tant que « Etudiant », je veux « accéder à mon espace de travail » pour « consulter les cours et l'état d'avancement des cours »
Estimation : 3 Points	Priorité : 1

- ♣ ID : un identifiant unique ;
- ♣ Scenario : description de la fonctionnalité attendue par le client ;
- ♣ Estimation : la quantité de travail nécessaire pour développer, tester et valider les stories
- ♣ Priorité : un entier qui fixe la priorité des stories ;

Au début d'un sprint l'équipe est réunie (développeurs et client) afin de déterminer quelles user stories vont être développées. Elles sont chiffrées lors d'une réunion intitulée « **sprint planning meeting** » et priorisées. L'ensemble des user stories sélectionnées constitue le « **sprint backlog** ». Le planning sprint et les objectifs à atteindre sont alors fixés.

Pour un logiciel de gestion et suivi des cours, un exemple de backlog de produit pourrait contenir les user stories « consulter les cours », « consulter l'état d'avancement des cours ».

Durant toute la durée du sprint, des réunions quotidiennes ou « **daily meetings** » sont organisées. Elles ont généralement lieu le matin et réunissent l'ensemble de l'équipe. Pour être efficace, une réunion quotidienne ne dure que 15 minutes et tous les participants restent debout. C'est un moyen particulièrement efficace pour s'assurer qu'elle ne s'éternise pas. L'objectif de cette réunion est de synchroniser l'équipe de façon à ce que chacun ait le même niveau d'information. Chaque développeur prend la parole à tour de rôle et décrit au reste de l'équipe ce qu'il a fait la veille, les objectifs qui ont été atteints, ce qu'il prévoit de faire aujourd'hui avec les nouveaux objectifs à atteindre, et les éventuels problèmes ou blocages qu'il rencontre. À la fin de la réunion, l'équipe valide avec le **Scrum Master** la possibilité de tenir les délais et de réaliser l'intégralité du sprint backlog.

À la fin du sprint, une démonstration de l'application dans son état actuel est faite au client. Le client comme l'utilisateur final peuvent manipuler l'application et vérifier que les développements effectués lors du sprint sont bien conformes à ce qui était attendu. Des remarques peuvent être émises et des demandes de modifications formulées. Elles pourront être ou non prises en compte dans le prochain sprint.

Lors du dernier jour de l'itération, une réunion appelée « **revue de sprint** » est également organisée. C'est le moment pour toute l'équipe de rappeler les objectifs qui ont été fixés et les fonctionnalités qui ont été réalisées et livrées.

Enfin, l'équipe projet se réunit une dernière fois durant le sprint pour la **rétrospective**. C'est l'occasion de faire la liste des processus qui ont bien fonctionné durant le sprint et de ceux qui nécessitent d'être améliorés. Tout le monde a la possibilité de s'exprimer librement. L'objectif est d'identifier les points forts et les points faibles du sprint dans un souci d'amélioration continue et de prendre en compte les différentes remarques lors des prochains sprints. Un plan d'amélioration est alors mis en place et le sprint suivant peut démarrer. La figure 1 illustre le fonctionnement de Scrum.

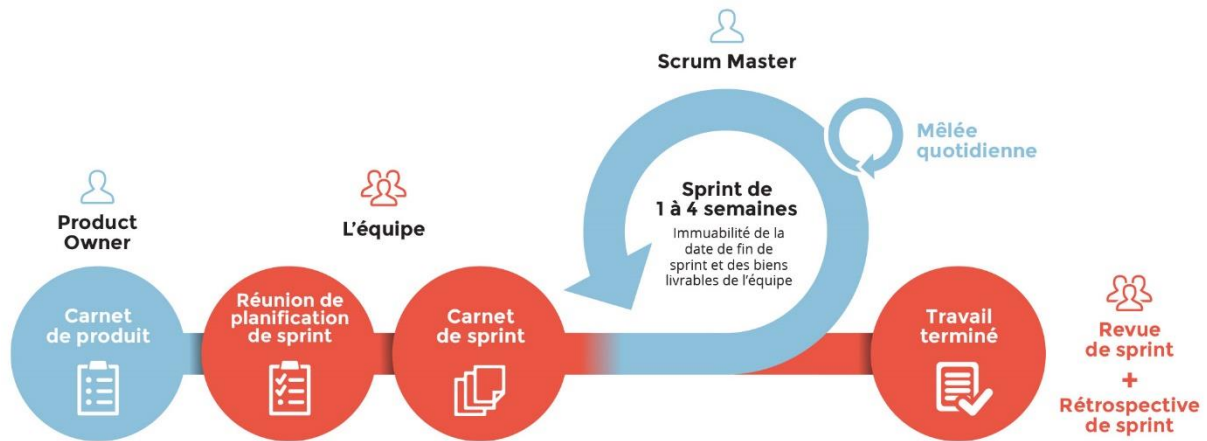


Figure 1 : Fonctionnement de Scrum

2.1.1. Les différents rôles de la méthodologie Scrum

Dans l'organisation d'un projet agile, Scrum définit trois rôles :

- ♣ **Le « Product Owner »** : il porte la vision du produit à réaliser et il s'agit donc généralement d'un expert métier. Il travaille en collaboration directe avec l'équipe de développement et a notamment la charge de remplir le **backlog** de produit et de déterminer la priorité des user stories à réaliser. Il peut être interne ou externe, même s'il s'agit généralement du client.
- ♣ **Le « Scrum Master »** : il s'agit d'un membre qui doit maîtriser Scrum car il est chargé de s'assurer que la méthodologie est correctement appliquée. Son rôle n'est pas de diriger, mais de faciliter le dialogue et le travail entre les différents intervenants, de façon à ce que l'équipe soit pleinement productive. Il doit être un bon communicant et faire preuve de pédagogie, afin de pouvoir résoudre les éventuels conflits qui pourraient apparaître durant le projet. Il anime généralement les différentes réunions, qu'il s'agisse des réunions quotidiennes, de la revue de sprint ou encore de la rétrospective.
- ♣ **L'équipe de développement** : généralement composée de 4 à 6 personnes, elle est chargée de transformer les besoins exprimés par le product owner sous la forme de user stories en fonctionnalités réelles, opérationnelles et utilisables. L'équipe est généralement composée de plusieurs profils, ne se limitant pas à des développeurs, et peut intégrer des architectes, un DBA (administrateur de base de données), un graphiste, un ergonomiste ou encore un ingénieur système ou réseau, en fonction des besoins.

2.1.2. Justification du choix sur Scrum

Le choix porté sur Scrum repose sur de nombreux avantages parmi lesquels nous pouvons citer :

- ♣ Implication du client dans le processus de développement ;
- ♣ Plus grande autonomie des développeurs ;
- ♣ Meilleure collaboration entre les équipes ;
- ♣ Meilleure gestion globale des risques.

2.1.3. Organisation de l'équipe

Dans cette partie, nous souhaitons appliquer la méthodologie de Scrum à notre projet. Par ailleurs, nous rappelons que cette méthodologie n'a pas été adoptée entièrement, mais nous avons seulement fait appel à ses pratiques. Le tableau ci-dessous représente l'équipe chargée de la réalisation du projet.

Tableau 2 : Présentation de l'équipe projet

Personnages	Rôles	Fonctions
Dr. Thierno Ahmadou DIALLO	Product Owner	Enseignant-chercheur au Département Informatique
M. Moussa TOURE	Scrum Master	Etudiant en Master 2 Génie logiciel
M. Moussa TOURE	Equipe	Etudiant en Master 2 Génie logiciel

2.2. Conclusion

À ce niveau, nous avons pu décrire Scrum et son mode de fonctionnement. À travers des arguments clairs et précis nous avons justifié notre choix sur cette méthode Agile. Ainsi, nous essayerons d'adapter ce processus de développement tout au long du projet, même si toutes les conditions ne sont pas réunies. Nous pouvons maintenant entamer un nouveau chapitre consacré à la spécification et à l'analyse des besoins fonctionnels.

Chapitre 3 Spécification et l'analyse des besoins fonctionnels

Dans ce chapitre, nous allons d'abord identifier les fonctionnalités et acteurs du système, ensuite nous allons montrer les diagrammes de cas d'utilisation et enfin nous terminons par la description des fonctionnalités.

3.1. Spécification des besoins fonctionnels

3.1.1. Identification des acteurs du système

En UML, un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié [2]. Les acteurs apparaissent dans les diagrammes de cas d'utilisation. L'acteur est toujours externe au système étudié. Un acteur est généralement représenté à l'aide d'une bonhomme allumette au-dessus ou en dessous duquel est écrit son nom, comme l'illustre la figure 2. Puisque la gestion et le suivi des cours au sein de l'UASZ est un travail qui nécessite un effort collectif. Ainsi, le cahier de texte électronique fait appel à six utilisateurs avec des profils différents (voir tableau 3).

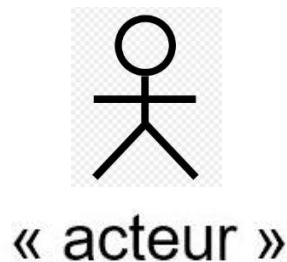


Figure 2 : Exemple de représentation d'un acteur

Tableau 3 : Identification des acteurs

<u>Acteurs</u>	<u>Actions</u>
Chef de département	Gestion des cours et du cahier de texte
Responsable de formation	
Responsable pédagogique	Gestion des services pédagogiques
Enseignant	Gestion du cahier de texte
Etudiant	
Administrateur	Administration du système

3.1.2. Identification des fonctionnalités du système

Les besoins fonctionnels expriment les fonctionnalités (actions exécutées) concrètes du système. Ce tableau ci-dessous, nous montre la répartition des fonctionnalités du système par acteurs.

Tableau 4 : Identification des fonctionnalités

<u>Fonctionnalités</u>	<u>Acteurs</u>
1. S'authentifier	- Chef de département ; - Responsable de formation; - Responsable pédagogique ; - Enseignant ; - Etudiant ; - Administrateur ;
2. Affecter des matières (modifier et supprimer affectations) aux enseignants	- Chef de département ; - Responsable de formation ;
3. Visualiser les enseignants	- Administrateur ; - Chef de département ; - Responsable de formation ;
4. Gérer les étudiants (ajouter, modifier, supprimer)	- Responsable pédagogique ;
5. Gérer les niveaux ou classe (ajouter, modifier, supprimer)	- Responsable pédagogique ;
6. Gérer les salles (ajouter, modifier, supprimer)	- Responsable pédagogique ;
7. Nommer le responsable de la classe	- Responsable pédagogique ;

8. Gérer les utilisateurs (ajouter, modifier, supprimer)	- Administrateur ;
9. Gérer les UFR (ajouter, modifier, supprimer)	- Administrateur ;
10. Gérer les départements (ajouter, modifier, supprimer)	- Administrateur ;
11. Gérer les formations (ajouter, modifier, supprimer)	- Administrateur ;
12. Gérer les enseignants (ajouter, modifier, supprimer)	- Administrateur ;
13. Consulter le tableau de bord	- Administrateur ; - Chef de département ; - Responsable de formation ; - Responsable pédagogique - Enseignant ; - Etudiant ;
14. Consulter les cours	- Chef de département ; - Responsable de formation; - Enseignant ; - Etudiant ;
15. Consulter l'état d'avancement des cours	- Chef de département ; - Responsable de formation; - Enseignant ; - Etudiant ;
16. Télécharger les supports de cours	- Etudiant ; - Enseignant ;
17. Activer le compte des étudiants	- Chef de département ; - Responsable de formation ;

<p>18. Visualiser les étudiants</p>	<p>- Enseignant ; - Chef de département ; - Responsable de formation ; - Responsable pédagogique ;</p>
<p>19. Exporter les données des étudiants</p>	<p>- Enseignant ;</p>
<p>20. Envoyer mail aux étudiants</p>	<p>- Enseignant ;</p>
<p>21. Imprimer les fiches de déclaration des heures de cours</p>	<p>- Enseignant ;</p>
<p>22. Modifier les informations personnelles</p>	<p>- Administrateur - Chef de département ; - Responsable de formation ; - Responsable pédagogique ; - Enseignant ; - Etudiant ;</p>

3.1.3. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation permettent de décrire les grandes fonctionnalités du système du point de vue des utilisateurs [2]. Un cas d'utilisation est un service rendu par le système et est souvent organisé en package. Les cas d'utilisation de notre système sont organisés en quatre packages :

- ♣ Administration du système (voir figure 3) ;
- ♣ Gestion des services pédagogiques (voir figure 4) ;
- ♣ Gestion des cours et du cahier de texte (voir figure 5) ;
- ♣ Gestion du cahier de texte (voir figure 6 et 7).

3.1.3.1. Diagramme de cas d'utilisation pour l'administration du système

L'administrateur assure la gestion des utilisateurs, des enseignants, des formations, des UFR ainsi que des départements. Il peut entre autres modifier ses informations personnelles et consulter son tableau de bord. Toutefois, il a l'obligation de s'authentifier avant d'effectuer une quelconque opération. La figure 3 illustre cette partie.

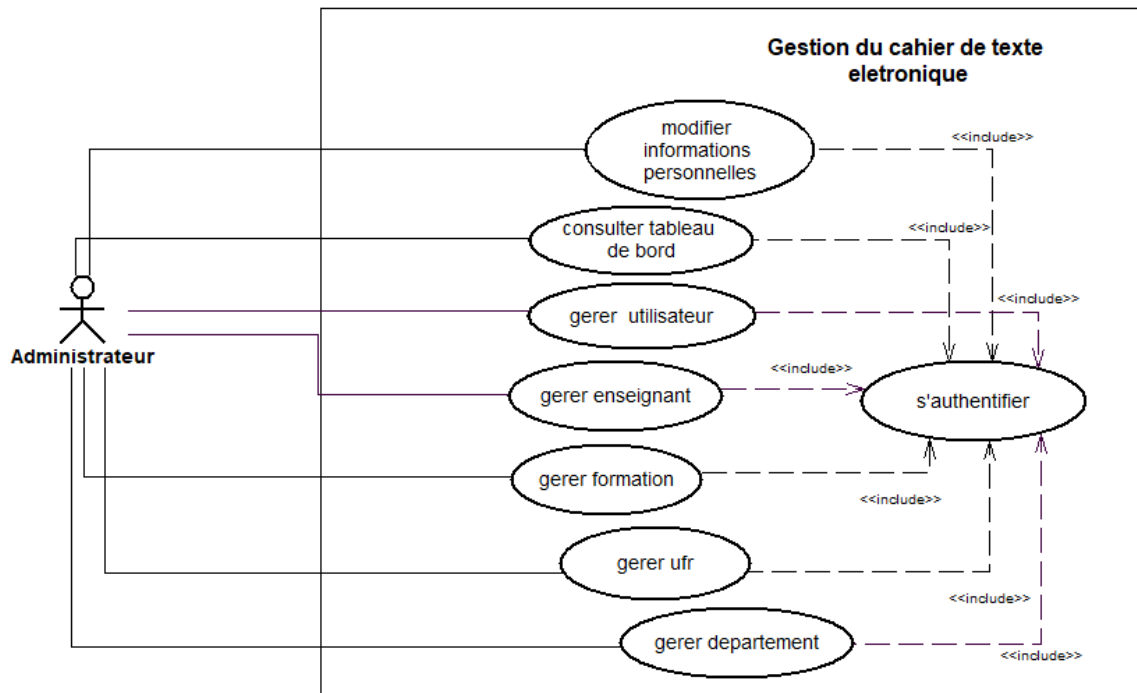


Figure 3 : Diagramme de cas d'utilisation de l'administrateur

3.1.3.2. Diagramme de cas d'utilisation pour la gestion des services pédagogiques

Le responsable pédagogique assure la gestion des salles, des niveaux et des étudiants. Il peut aussi modifier ses informations personnelles, nommer le responsable de la classe, de même que consulter son tableau de bord. Cependant, une authentification est nécessaire au préalable. La figure 4 illustre cette partie.

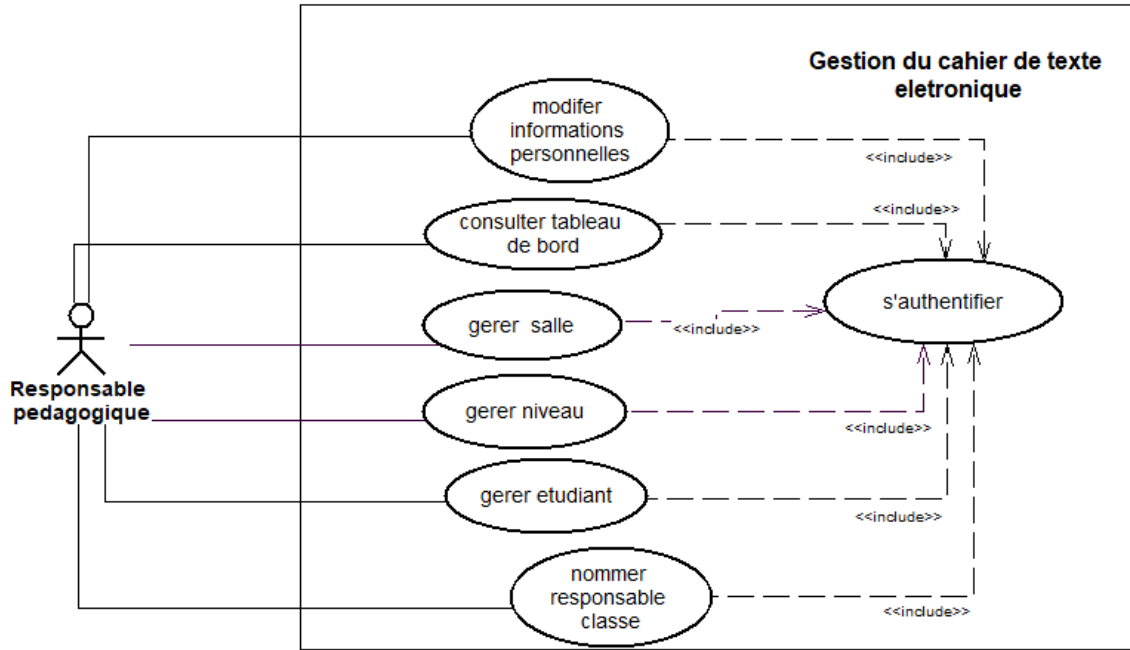


Figure 4 : Diagramme de cas d'utilisation des responsables pédagogiques

3.1.3.3. Diagramme de cas d'utilisation pour la gestion des cours et du cahier de texte

Après authentification, le chef de département comme le responsable de formation ont comme tâche : la gestion des cours et du cahier de texte. De façon explicite, ils peuvent affecter des matières aux enseignants, modifier une affectation ou de la supprimer, consulter leur tableau de bord, visualiser les enseignants, consulter les cours et leur état d'avancement, visualiser les étudiants et enfin d'activer le compte des étudiants. La figure 5 illustre cette partie.

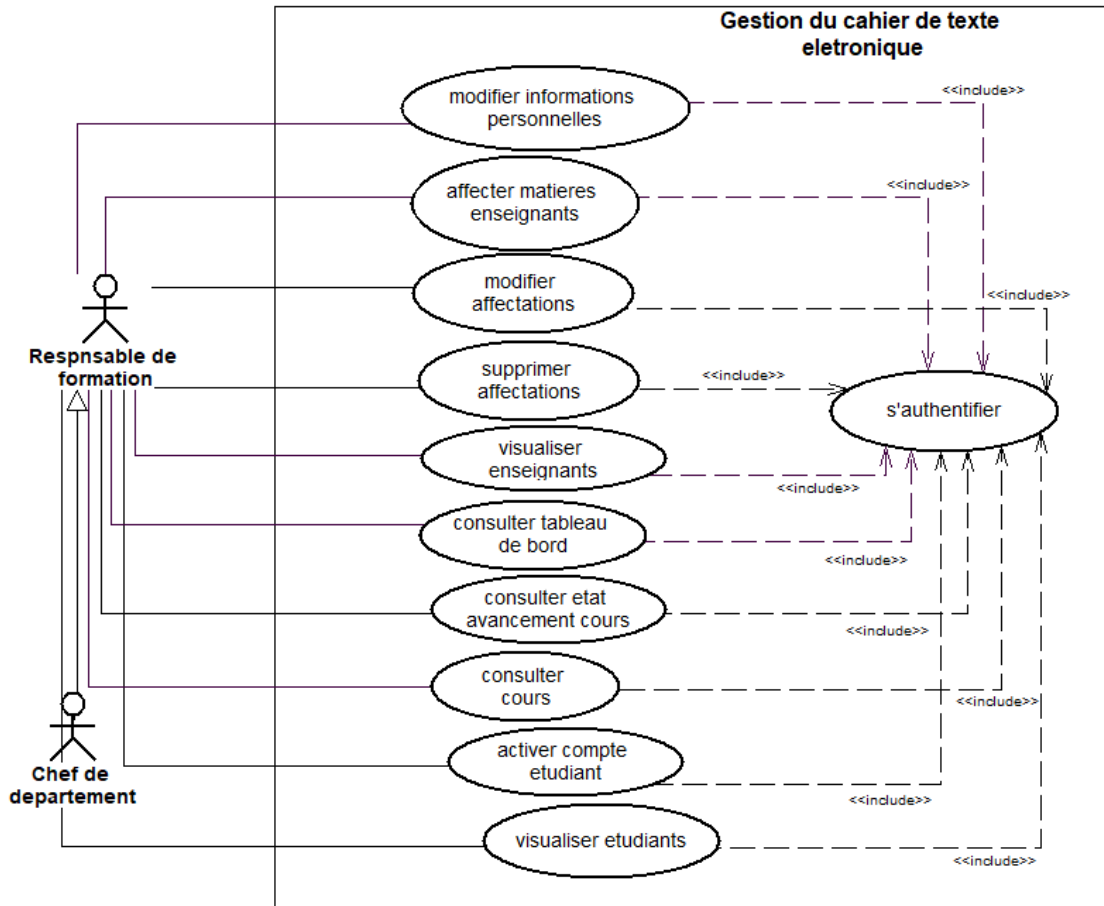


Figure 5 : Diagramme de cas d'utilisation des responsables de formation et chefs de département

3.1.3.4. Diagramme de cas d'utilisation pour la gestion du cahier de texte

Après avoir rempli les conditions d'authentification, l'étudiant peut modifier ses informations personnelles, consulter son tableau de bord, consulter les cours et leur état d'avancement, télécharger le support de cours et enfin remplir le cahier de texte. La figure 6 illustre cette partie.

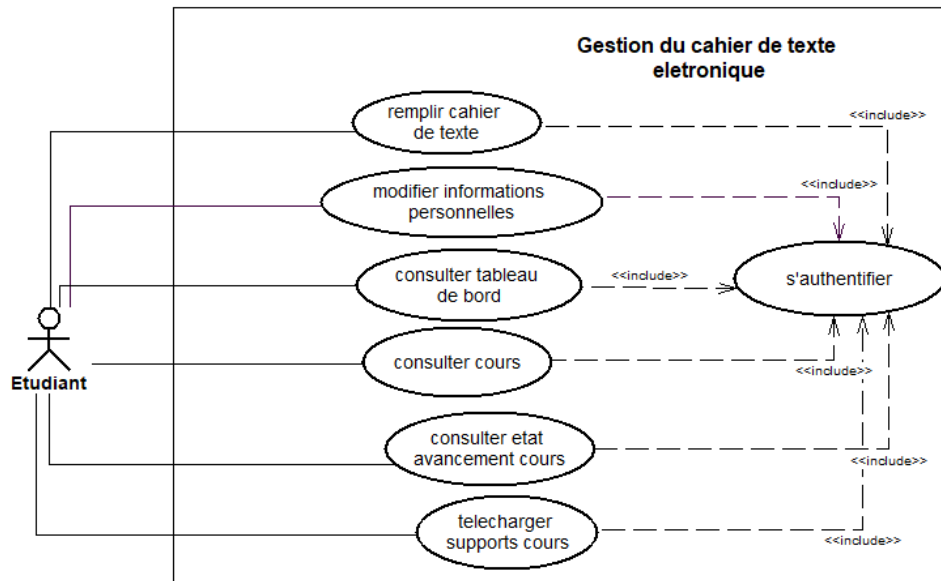


Figure 6 : Diagramme de cas d'utilisation de l'étudiant

Après avoir franchi l'étape d'authentification, l'enseignant dispose certains droits tel que : modifier ses informations personnelles, consulter son tableau de bord, consulter les cours et leur état d'avancement, télécharger le support de cours, valider le contenu du cahier de texte, visualiser les étudiants, d'envoyer des mails aux étudiants, d'exporter les données des étudiants et enfin d'imprimer des fiches de déclarations des heures de cours. La figure 7 illustre cette partie.

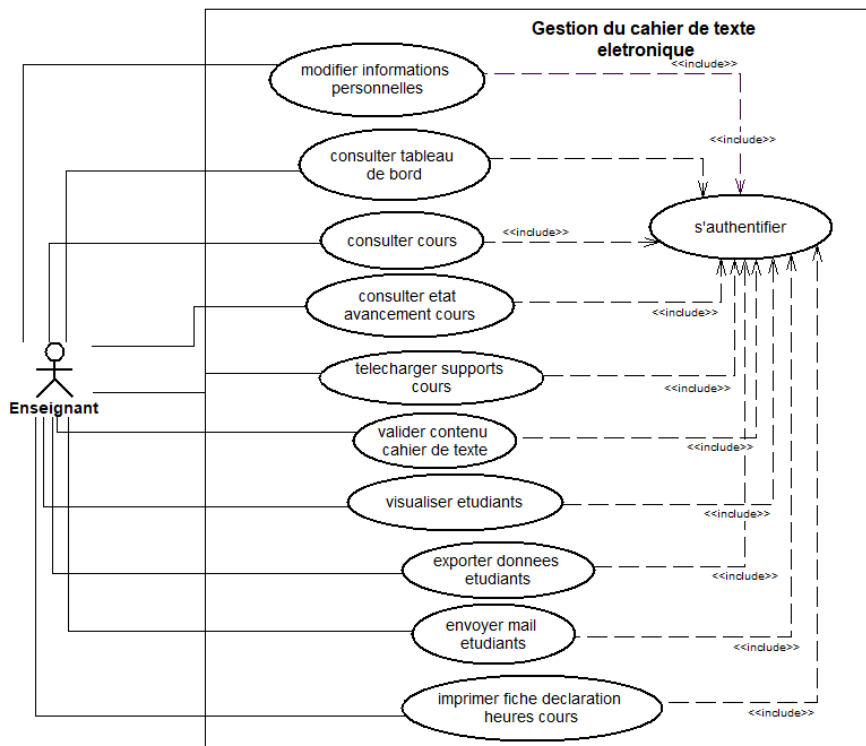


Figure 7 : Diagramme de cas d'utilisation de l'enseignant

3.2. Analyse des besoins fonctionnels du système

3.2.1. Analyse de l'authentification

Dans cette partie, nous allons d'abord décrire le cas d'utilisation « s'authentifier » avant de passer à l'élaboration du diagramme d'activité et de séquence.

3.2.1.1. Description du cas d'utilisation « s'authentifier »

Il s'agit de reprendre le cas d'utilisation « s'authentifier » et de le décrire très précisément, tel qu'illustré par la figure ci-dessous.

Tableau 5 : Description du cas d'utilisation « s'authentifier »

Sommaire d'identification
Titre : s'authentifier
Objectif : gérer l'authentification des utilisateurs.
Résumé : cette fonctionnalité permet d'identifier les utilisateurs du système ainsi que leurs espaces de travail.
Acteur : Administrateur, Chef de département, Responsable de formation, Responsable pédagogique, Enseignant, Etudiant.
Description détaillée : Pré conditions : avoir un compte d'utilisateur Description du traitement nominal : <ol style="list-style-type: none"> 1. L'acteur saisit son identifiant et son mot de passe ; 2. Le système vérifie les informations saisies ; 3. Le système récupère le profil de l'utilisateur ;
Post conditions : accéder à son espace de travail.
Exceptions : [Exception 1 : ChampsObligatoires] : Message d'erreur si l'un des champs obligatoire n'est pas rempli. [Exception 2 : IdentifiantMotDePasseIncorrecte] : Message d'erreur si l'identifiant ou le mot de passe est incorrect.

3.2.1.2. Diagramme de séquence du cas d'utilisation « s'authentifier »

Le diagramme de séquence permet de décrire les scénarios de chaque cas d'utilisation [2]. Il permet aussi de représenter des interactions entre objets selon un point de vue temporel.

Lorsque l'utilisateur souhaite accéder à son espace de travail, une page d'authentification lui sera affichée dès le démarrage de l'application, dans laquelle il saisit ses propres coordonnées d'authentification (login et mot de passe). Par la suite, le système procède à la vérification des informations introduites, si le login et le mot de passe sont valides son espace de travail lui sera alors ouvert, sinon un message d'erreur est affiché le sollicitant de réintroduire ses coordonnées. Ce processus de vérification se répétera autant de fois que l'utilisateur communique des informations erronées. Cette image ci-dessous, nous montre les scénarios du cas d'utilisation de l'authentification.

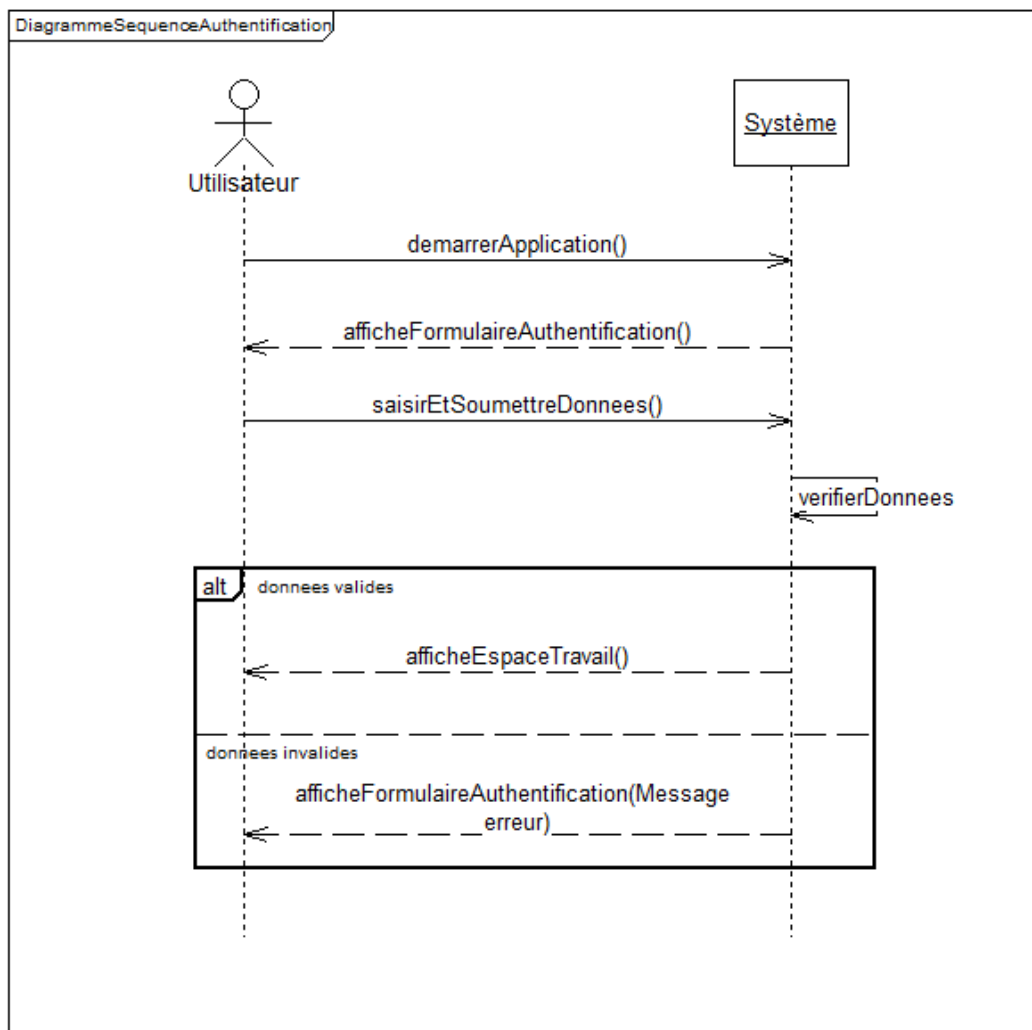


Figure 8 : Diagramme de séquence du cas d'utilisation « s'authentifier »

3.2.1.3. Diagramme d'activité du cas d'utilisation « s'authentifier »

Le diagramme d'activité montre l'enchaînement des actions et des décisions au sein d'une activité du système, ou au sein de tout le système [2]. Une activité représente un déroulement d'étapes séquentielles.

Le cas d'utilisation « s'authentifier » regroupe plusieurs activités. L'activité démarre par le démarrage de l'application qui coïncide avec la page d'authentification. Ensuite, l'utilisateur saisit les coordonnées d'authentification, le système procède à la vérification des informations introduites et prend des décisions. Si le login et le mot de passe sont valides alors le système déclenchera l'activité « affiche espace de travail ». Dans le cas contraire (login et le mot de passe sont invalides), l'activité « affiche message erreur » est ouverte sollicitant l'utilisateur de réintroduire ses coordonnées. Ce processus de vérification se répétera autant de fois que l'utilisateur communique des informations erronées, comme l'illustre la figure ci-dessous.

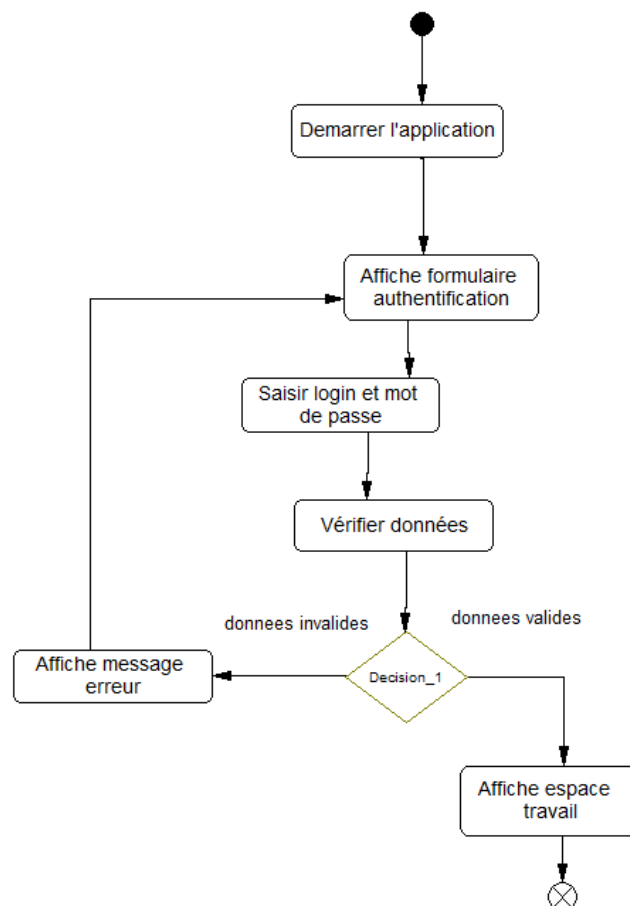


Figure 9 : Diagramme d'activité du cas d'utilisation « s'authentifier »

3.2.2. Analyse de la gestion administrative du système

Dans cette section, nous allons d'abord faire une description du cas d'utilisation « ajouter utilisateur » ensuite nous terminons par une illustration du diagramme d'activité et de séquence.

3.2.2.1. Description du cas d'utilisation « ajout utilisateur »

Il s'agit de reprendre le cas d'utilisation « ajout utilisateur » et de le décrire très précisément, tel qu'illustré par la figure ci-dessous.

Tableau 6 : Description du cas d'utilisation « gérer utilisateur »

Sommaire d'identification
Titre : gérer utilisateurs
Objectif : création des comptes utilisateurs
Résumé : cette fonctionnalité permet à l'administrateur d'ajouter, de modifier et de supprimer un utilisateur.
Acteur : Administrateur
Description détaillée : Pré conditions : l'administrateur s'est authentifié sur le système Description du traitement nominal :
L'acteur peut : <ol style="list-style-type: none"> 1. Ajouter un utilisateur ; 2. Modifier un utilisateur ; 3. Supprimer un utilisateur ;
Post conditions : gérer les utilisateurs du système.
Exceptions : [Exception 1 : ChampsObligatoires] : Message d'erreur si l'un des champs n'est pas rempli.

3.2.2.2. Diagramme de séquence du cas d'utilisation « ajout utilisateur »

Une fois que l'administrateur accède à son espace de travail et qu'il souhaite ajouter un utilisateur. Un formulaire ajout utilisateur lui sera affiché, dans laquelle il renseigne toutes les informations. Par la suite, le système procède à la vérification des informations introduites, s'ils sont valides le système sauvegardera les informations, sinon un message d'erreur est affiché le sollicitant de ressaisir les informations. Ce processus de vérification se répétera autant de fois

que l'utilisateur communique des informations erronées. La figure 10 décrit le diagramme de séquence du cas d'utilisation « ajout utilisateur ».

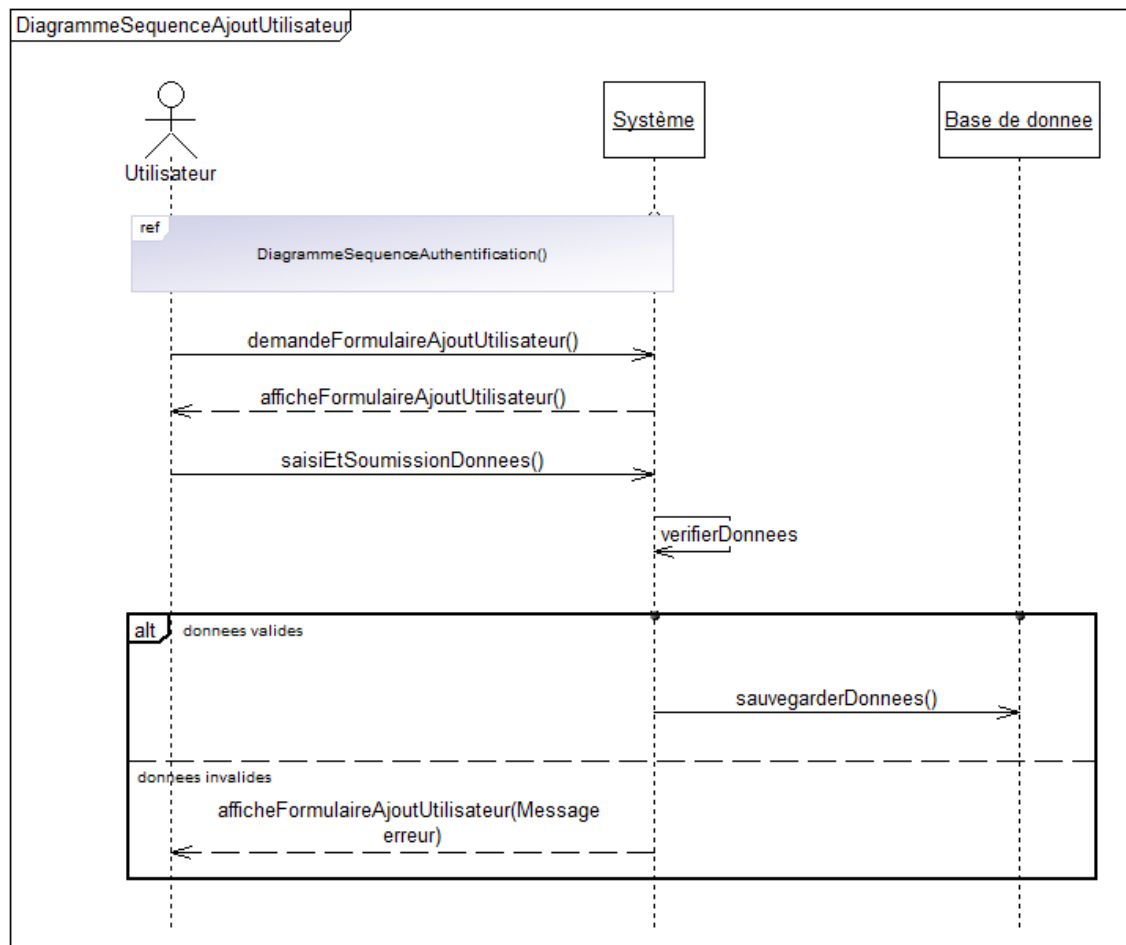


Figure 10 : Diagramme de séquence du cas d'utilisation « ajout utilisateur »

3.2.2.3. Diagramme d'activité du cas d'utilisation « ajout utilisateur »

Le cas d'utilisation « ajout utilisateur » regroupe plusieurs activités. L'activité démarre au préalable par une authentification. Lorsque l'administrateur souhaite ajouter un utilisateur, un formulaire ajout utilisateur lui sera affiché. Ensuite, l'utilisateur renseigne les informations, le système procède à la vérification des informations introduites et prend des décisions. Si les données sont valides alors l'activité « sauvegarderDonnees » sera activée. Sinon (si les données sont invalides) alors l'activité « affiche message d'erreur » est ouverte le sollicitant de ressaisir les informations. Ce processus de vérification se répétera autant de fois que l'utilisateur communique des informations erronées, comme l'illustre la figure ci-dessous.

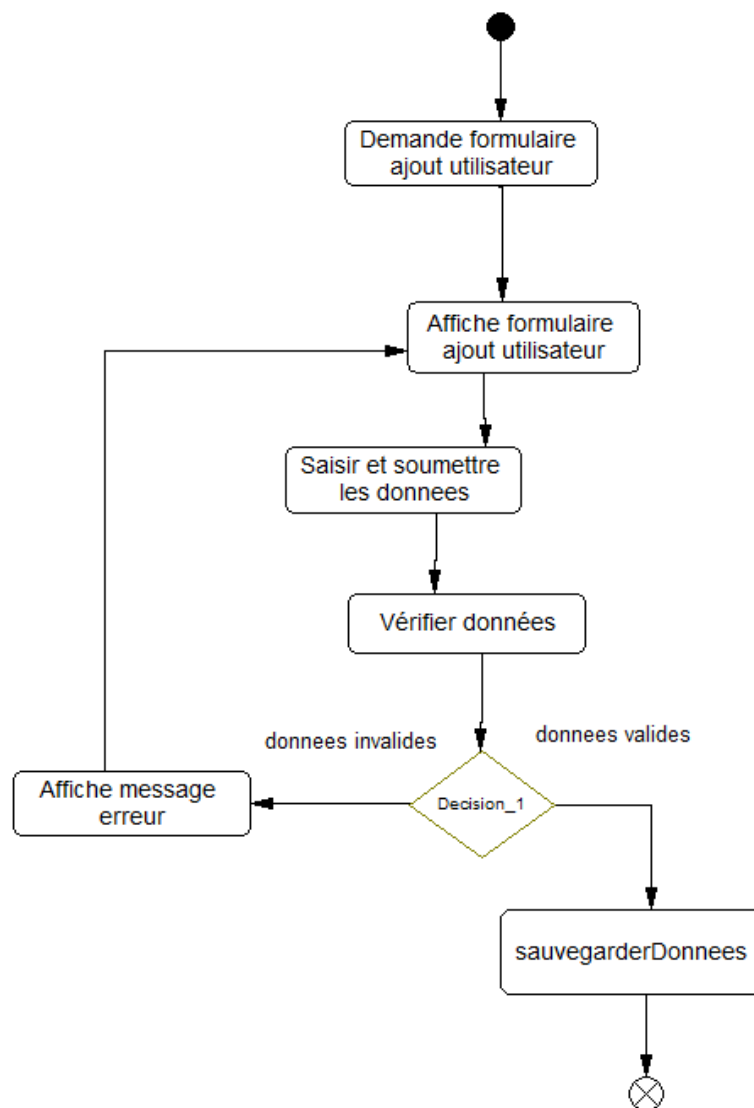


Figure 11 : Diagramme d'activité du cas d'utilisation « ajout utilisateur »

3.3. Conclusion

À l'issue de cette phase, nous avons pu exprimer clairement la spécification et l'analyse des besoins fonctionnels du futur système. Nous pouvons maintenant introduire le chapitre sur la conception du système.

Chapitre 4 Conception du système

La conception de logiciel est un art qui nécessite de l'expérience [19]. Elle consiste à traduire les besoins en spécifiant comment l'application pourra satisfaire les utilisateurs avant de procéder à sa réalisation. Dans ce chapitre, nous décrivons les différentes étapes de la réalisation de notre application en commençant par la conception générale et en terminant par la conception détaillée.

4.1. Conception générale

4.1.1. Architecture de l'application

L'architecture d'une application décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. En ce qui concerne notre application, nous adaptons l'architecture 3 - tiers illustrée par la figure 12 et organisée comme suit :

- ♣ **Couche présentation** : pour la partie présentation de l'application ;
- ♣ **Couche métier** : pour le traitement de l'information ;
- ♣ **Couche accès aux données** : pour l'accès et stockage des données.

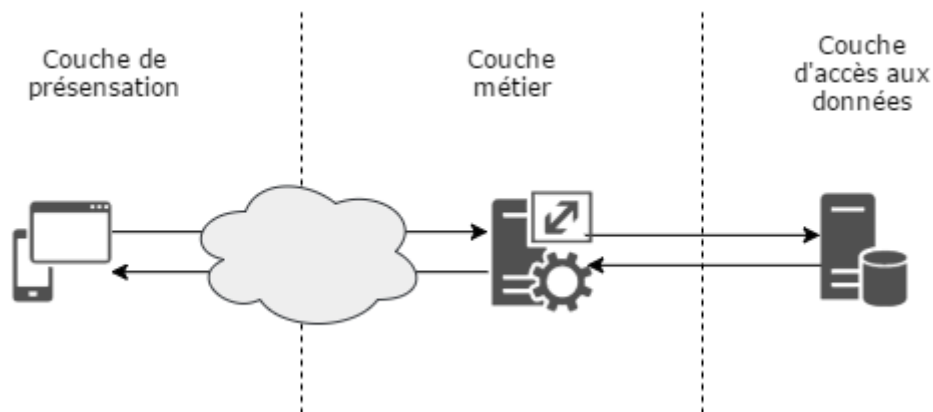


Figure 12 : Architecture de l'application

Au niveau de la couche métier, nous avons appliqué un modèle de développement connu sous le nom de MVC (modèle - vue - contrôleur). Ce modèle sépare la couche interface utilisateur des autres parties du système et est composé de trois types de composants que sont :

- ♣ **Modèle** : contient les données à afficher ;
- ♣ **Vue** : utilisé pour présenter et afficher les données du modèle dans l'interface utilisateur ;

- ♣ Contrôleur : contient les fonctionnalités nécessaires pour gérer et contrôler les interactions de l'utilisateur avec la vue et le modèle.

4.1.2. Diagramme de composant

Également connus sous le nom de diagramme de flux des composants, il montre les groupes logiques d'éléments et leurs relations. En d'autres termes, il offre une vue plus simplifiée d'un système complexe en le désossant en plus petits composants. Chacun des éléments est représenté par une case rectangulaire comportant son nom. Les connecteurs définissent la relation et les dépendances entre les différents composants, comme l'illustre la figure 13.

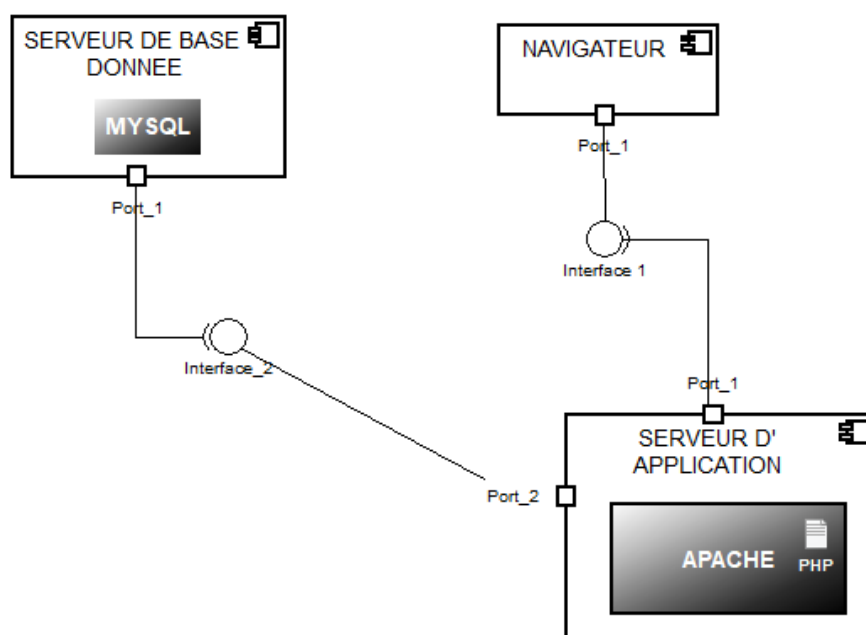


Figure 13 : Diagramme de composant

4.1.3. Diagramme de package

Il est utilisé pour représenter les dépendances entre les paquetages qui composent un modèle. L'objectif principal est de montrer la relation entre les différents grands composants qui forment un système complexe. La figure 14 décrit le diagramme de package de notre système.

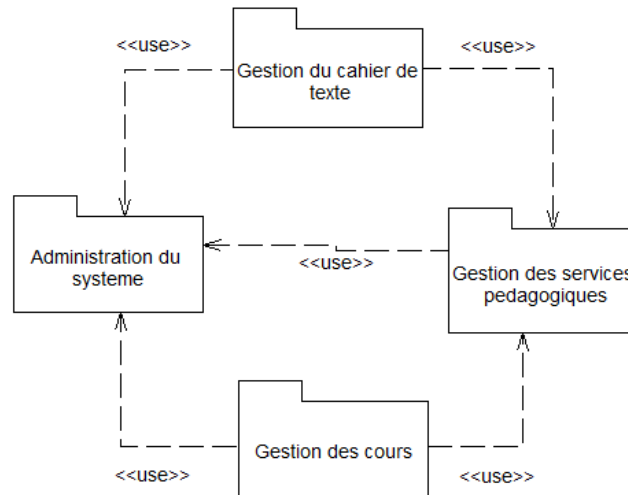


Figure 14 : Diagramme de package

4.1.4. Diagramme de déploiement

Il affiche les composants matériels et logiciels ainsi que leurs relations. Il apporte une représentation visuelle de l'endroit exact où chaque composant logiciel est déployé, comme illustré par la figure 15.

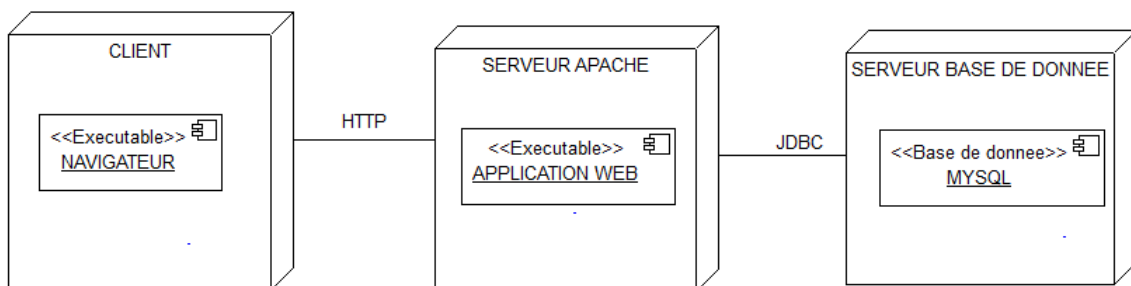


Figure 15 : Diagramme de déploiement

4.2. Conception détaillée

4.2.1. Dictionnaire des données

La collection et l'analyse des informations en provenance de différentes sources, nous a permis d'établir le dictionnaire de données en annexe A de ce document.

4.2.2. Diagramme de classe

Le diagramme de classe est une représentation statique du système à modéliser. Il montre la passation des différentes opérations. L'analyse systématique des données du dictionnaire permet de les regrouper dans des entités à part et de les représenter de la façon suivante.

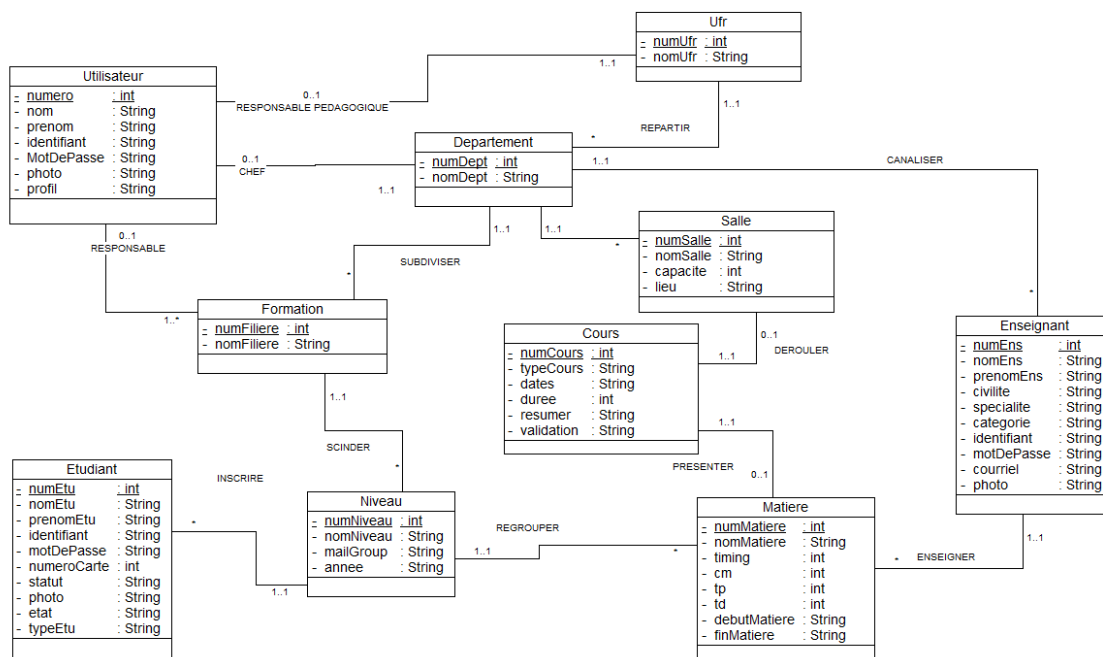


Figure 16 : Diagramme de classe

Ce diagramme est reparti en dix classes. Un utilisateur est soit un responsable pédagogique d'une UFR, soit le chef d'un département ou le responsable d'une formation. Une formation est scindée en plusieurs niveaux. Dans chaque niveau, un nombre important d'étudiant y sont inscrit. Chaque niveau regroupe un ensemble de matière. Dans un cours, une seule matière est présentée et le cours se déroule dans une salle. La salle appartient à un département et chaque département a la possibilité de recruter plusieurs enseignants. L'enseignant a le privilège d'enseigner plusieurs matières.

4.2.3. Diagramme de classe participante à la gestion de l'authentification des utilisateurs

Conçu pour gérer l'authentification des utilisateurs ainsi que leur espace de travail. En pratique, lors de l'ouverture d'une session (authentification), le système fait appel à une liste de classe (utilisateur, UFR, département et formation) pour assurer l'authentification, tel qu'illustré dans la figure ci-dessous.

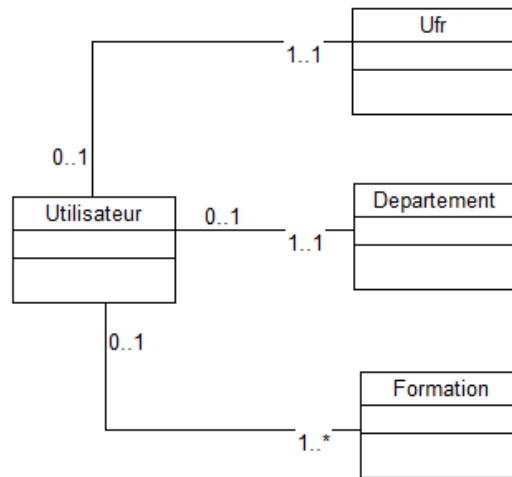


Figure 17 : Diagramme de classe participante à l'authentification des utilisateurs

4.2.4. Diagramme de classe participante à la gestion des cours

Ce diagramme permet de représenter les entités qui interviennent dans la gestion des cours. A travers ce diagramme, lorsque le chef de département ou le responsable de formation souhaite par exemple affecter des matières aux enseignants, le système fait appel à une liste de classe (cours, matière, salle, enseignant, département) pour effectuer l'opération demandée. La figure 18 décrit le diagramme de classe participante à la gestion des cours.

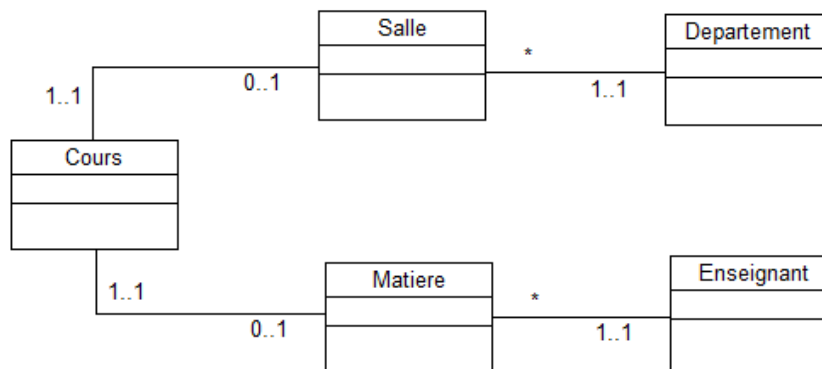


Figure 18 : Diagramme de classe participante à la gestion des cours

4.3. Conclusion

Dans ce chapitre, nous avons pu comprendre que le modèle d'architecture produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique, mais plutôt comment il doit être conçu de manière à répondre aux spécifications. Il faut noter que cette étape est une activité importante qui va nous permettre d'introduire le dernier chapitre : l'implémentation et la présentation de l'application.

Chapitre 5 Implémentation et présentation de l'application

Nous allons enfin entamer la dernière phase de notre travail par lister les technologies utilisées avant de montrer l'installation et la configuration du Framework CodeIgniter. Nous allons ensuite présenter le processus de l'implémentation de la base de données, puis de l'application et enfin nous terminons par la présentation des différentes interfaces de l'application.

5.1. Technologies utilisées

Dans cette partie, nous allons définir quelques généralités portant sur les méthodes et outils utilisés pour la réalisation de notre projet.

5.1.1. Xampp

C'est une plateforme de développement Web de type XAMPP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Xampp n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows). [10]

5.1.2. Power AMC

POWER AMC est l'un des premiers outils qui permet d'élaborer des modèles de données que cela soit **MERISE**, **UML** ou autre, de manière graphique et de les implémenter quel que soit le SGBD et de manière automatique. De même, l'outil permet de modéliser les processus métiers.

Aussi Power AMC est une force dans tout nouveau projet d'entreprise car il permet d'identifier avec précision quels processus, quelles personnes et/ou quelles données seront impactés. [2]

5.1.3. Uml

Unified Modeling Language est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un **formalisme graphique** issu de notations employées dans différentes méthodes objets.

En effet, UML utilise l'approche objet en présentant un langage de description universel. Il permet grâce à un ensemble de diagrammes très explicites, de représenter l'architecture et le

fonctionnement des systèmes informatiques complexes en tenant compte des relations entre les concepts utilisés et l'implémentation qui en découle.

Il représente aussi un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.

Il représente un juste milieu entre langage mathématique et naturel, pas trop complexe, mais suffisamment rigoureux, car basé sur un méta modèle. Une autre caractéristique importante d'UML, est qu'il cadre l'analyse. UML permet de représenter un système selon différentes vues complémentaires : les diagrammes (le diagramme de classe, le diagramme de cas d'utilisation, le diagramme de séquence etc...). [2]

5.1.4. JavaScript

JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le serveur), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au-dessus d'un élément, du redimensionnement de la page...). [6]

5.1.5. Bootstrap

Bootstrap est une compilation de plusieurs éléments et fonctions web-design personnalisables, le tout emballé dans un seul et même outil. Les développeurs qui utilisent Bootstrap pour la création de leur site web choisissent les éléments qu'ils veulent utiliser avec la certitude qu'ils ne seront pas incompatibles entre eux.

Les éléments personnalisables compilés dans Bootstrap sont une combinaison de HTML, CSS et JavaScript. Et grâce à la magie de l'open-source, Bootstrap s'améliore en permanence : de nouvelles fonctions absolument géniales ont été ajoutées comme le mobile responsive ou la très large sélection de plugins jQuery. [8]

5.1.6. HTML5 / CSS3

HTML est une abréviation de : L'**H**yper**T**ext **M**arkup **L**anguage (Langage de balisage d'hypertexte), langage de programmation pour les pages Internet. Ce langage informatique n'est pas réellement un langage de programmation, mais un langage de balisage. En d'autres mots c'est un format de données qui permet de concevoir une page web. Révélé en 1991 par Tim Berners-Lee, le HTML fait partie des bases de l'internet. Il découle directement du SGML.

Les feuilles de style (*Cascading Style Sheets*) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C) au même titre que HTML ou XML.

Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc.

Le but de CSS est de séparer la structure d'un document HTML et sa présentation. De plus, CSS ajoute des fonctionnalités nouvelles par rapport à HTML au point de vue du style. [1]

5.1.7. PHP

Le PHP est un langage informatique utilisé sur internet. Le terme PHP est un acronyme récursif de "PHP: HyperText Preprocessor".Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL.

Exécuté du côté serveur (l'endroit où est hébergé le site) il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier. Néanmoins, les webmasters qui souhaitent développer un site en PHP doivent s'assurer que l'hébergeur prend en compte ce langage.

Lorsqu'une page PHP est exécutée par le serveur, alors celui-ci renvoie généralement au client (aux visiteurs du site) une page web qui peut contenir du HTML, XHTML, CSS, JavaScript. [3,14]

5.1.8. Framework CodeIgniter

CodeIgniter est un environnement cadre de développement d'application, un ensemble d'outils permettant de structurer et de construire des sites Web en utilisant PHP. Son objectif est de vous permettre de développer des projets beaucoup plus rapidement que si vous partiez de zéro, en fournissant un ensemble fourni de bibliothèques pour les tâches habituellement nécessaires, ainsi qu'une interface simple et une structuration logique d'accès à ces bibliothèques.

CodeIgniter vous permet de vous concentrer sur votre créativité en minimisant la quantité de code nécessaire pour réaliser une tâche donnée.

CodeIgniter est une Framework simple et léger capable de vous permettre de réaliser un mini site rapidement en vous fournissant le cadre de développement favorable à une progression rapide allié à une structuration idéale de votre projet. [9,12]

5.1.9. TCPDF

TCPDF est une classe PHP, d'utilisation répandue, permettant de créer des documents PDF. Elle est libre et open source. TCPDF est actuellement la seule bibliothèque PHP de génération PDF offrant un support complet de l'encodage UTF-8 et des langues s'écrivant de droite à gauche, incluant le texte bidirectionnel.

Elle est intégrée dans des CMS, applications et Framework PHP répandus, par exemple : Joomla! 3, Drupal4, Moodle5, Prestashop, phpMyAdmin6, Symfony7. [20]

5.1.10. PHPExcel

PHPExcel est une bibliothèque open source qui permet de lire et d'écrire dans des tableurs, XLS et XLSX. Mais il peut aussi générer des CSV, des PDF, et des HTML.

Elle comprend toute sorte de fonctions de manipulations de tableurs, telles que le changement de couleur des champs, l'ajout de graphiques et de filtres, la protection de feuilles. [21]

5.1.11. Outil de développement de l'application : Netbeans

Netbeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, Netbeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons.

Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Compilé en Java, Netbeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

Netbeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). [15]

5.2. Implémentation

5.2.1. Installation et configuration du Framework CodeIgniter

5.2.1.1. Installation du Framework CodeIgniter

Il suffit juste de télécharger le fichier compressé puis de l'extraire dans le dossier « **htdocs** » de xampp ensuite le renommer, par exemple « Ctexte ». Nous pouvons vérifier si l'installation s'est passée correctement en tapant sur l'url <http://localhost/Ctexte/>. Normalement, la page d'accueil par défaut de CodeIgniter devrait apparaître et ressemble à l'image ci-dessous.

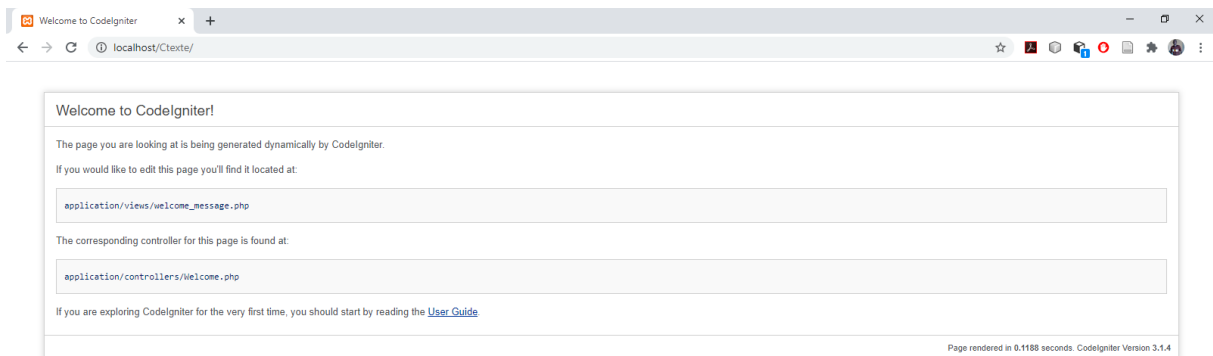


Figure 19 : Page d'accueil après l'installation de CodeIgniter

5.2.1.2. Configuration du Framework CodeIgniter

Après installation, nous constatons que CodeIgniter est structuré de la manière suivant :

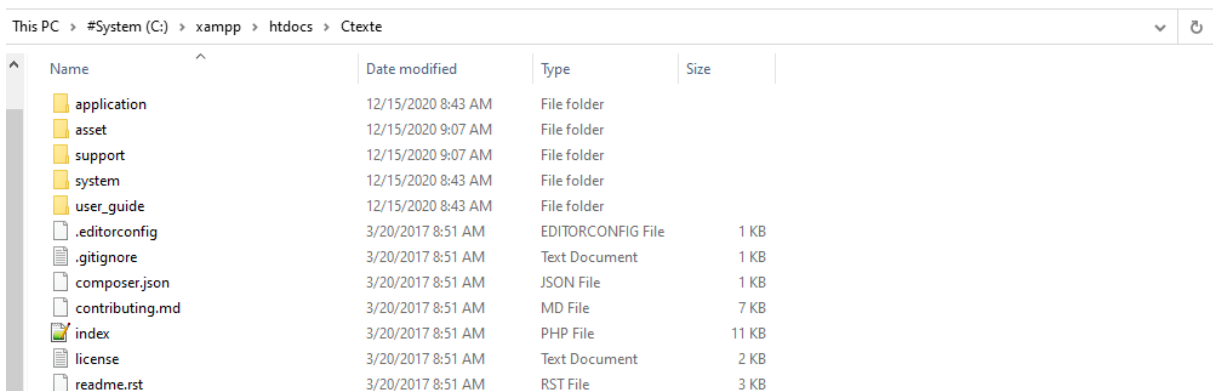


Figure 20 : Capture d'écran du contenu du répertoire de notre application

Pour bien décrire les composants de CodeIgniter nous allons nous focaliser en grande partie sur le dossier nommé application.

Il est constitué d'un ensemble de dossiers répartis comme suit :

- ♣ un dossier « **config** » : regroupe l'ensemble des fichiers de configuration de CodeIgniter. De manière effective, il représente le dossier de base. À l'intérieur, nous y trouvons :
 - ❖ Le fichier « **autoload.php** » : comme son nom l'indique le chargement se fait de manière automatique. Pour utiliser par exemple les sessions, il faut d'abord se positionner sur ce fichier, ensuite dans « **librairies** » puis ajouter le nom « **session** » ou « **database** » pour charger la base de données puis enregistrer. Il en est de même pour charger les modèles dans « **model** ».
 - ❖ Le fichier « **routes** » : définit le contrôleur à charger par défaut.
 - ❖ Le fichier « **database** » : contient toutes les informations liées à la base données.
- ♣ un dossier « **controllers** » permettant de répertorier l'ensemble des contrôleurs de l'application.
- ♣ un dossier « **models** » pour interroger la base de données.
- ♣ un dossier « **views** » contenant l'ensemble des pages web de l'application.

Remarque : nous avons la possibilité de créer autant de dossiers et de sous-dossiers dans le répertoire « Ctexte » à toutes fins utiles. Par exemple le dossier « **assets** » ou « **support** ». En outre, nous avons un dossier nommé « **user_guide** » qui sert comme guide d'utilisation pour CodeIgniter. Pour en savoir plus, lancer le fichier « **index.php** » qui se trouve dans le répertoire « **user_guide** ».

5.2.2. Implémentation de la base de données

5.2.2.1. Modèle physique de données

Générée automatiquement à partir du diagramme de classe, cette étape finalise le processus de traitement des données. La figure 21 illustre le modèle physique de donnée.

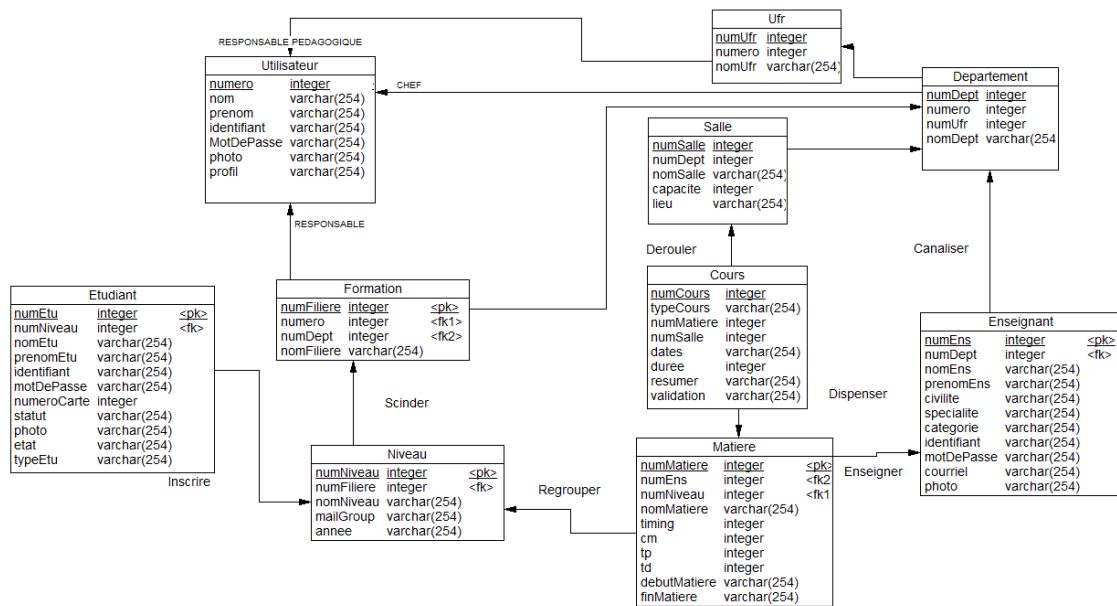


Figure 21 : Modèle physique des données

5.2.2.2. Création de la base de données sous MySQL

Après avoir installé Xampp, il nous faut créer à l'intérieur de MySQL une base de données. Pour configurer le serveur de bases de données, nous sommes passés par un utilitaire qui se nomme phpMyAdmin. Il se situe dans la liste des outils de Xampp. PhpMyAdmin est un programme écrit en PHP qui s'exécute à l'intérieur d'un navigateur Web.

a. Création de la base de données avec phpMyAdmin

Il est nécessaire de créer une base de données qui contiendra l'ensemble des fichiers de l'application. Cette base de données sera nommée « **ctexte** », comme l'illustre la figure 22.

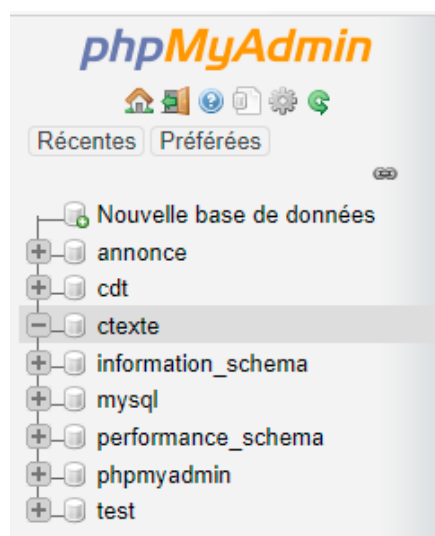


Figure 22 : Capture d'écran après la création de la base de données.

b. Importation de la base de données

Grâce au modèle physique de données, nous avons pu générer un fichier « **ctexte.sql** ». Ce dernier sera importé dans la base de données « **ctexte** » créée à partir de PhpMyAdmin. La procédure à suivre pour importer ce fichier est la suivante :

1. Sélectionner le nom de la base de données ;
2. Cliquer sur l'onglet importé ;
3. Ensuite cliquer sur « **parcours les fichiers** » pour sélectionner le fichier « **ctexte.sql** » ;
4. Terminer la partie en cliquant sur « **exécuter** »

La figure 23 et 24 est en parfaite corrélation de ce que nous venons de dire.

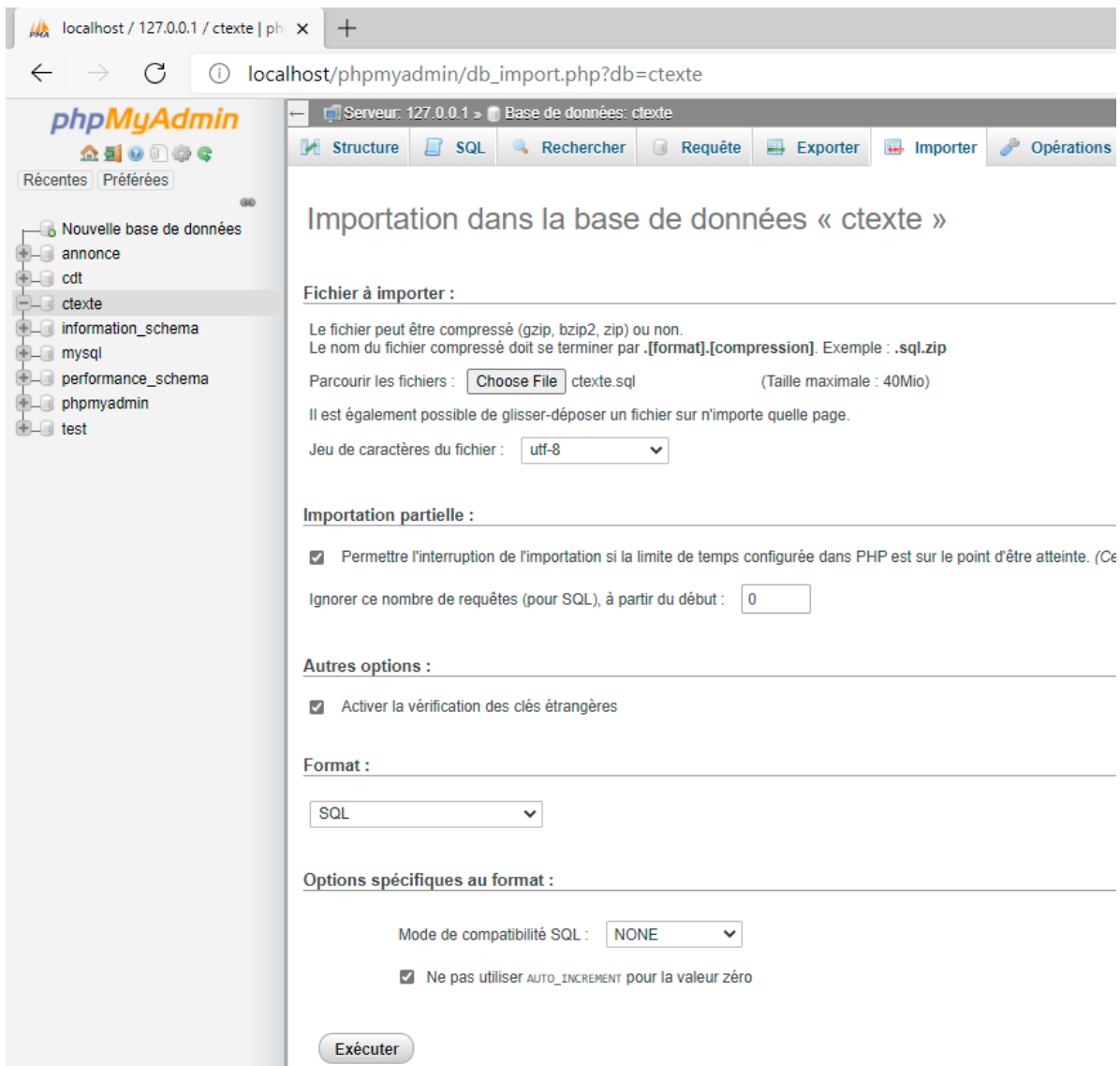


Figure 23 : Importation de la base de données.

c. Vue globale de la base de données

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> cours	★ Parcourir Structure Rechercher Insérer Vider Supprimer	37	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> departement	★ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> enseignant	★ Parcourir Structure Rechercher Insérer Vider Supprimer	8	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> etudiant	★ Parcourir Structure Rechercher Insérer Vider Supprimer	29	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> formation	★ Parcourir Structure Rechercher Insérer Vider Supprimer	8	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> matiere	★ Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> niveau	★ Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> salle	★ Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> ufr	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> utilisateur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	14	InnoDB	utf8_general_ci	16,0 kio	-
10 tables	Somme	133	InnoDB	utf8_general_ci	368,0 kio	0 0

Figure 24 : Capture d'écran des tables de la base de données

d. Etablissement de la connexion vers la base de données

Puisque nous avons utilisé le Framework CodeIgniter, il est impératif de passer par les fichiers de configuration de ce fichier pour pouvoir se connecter à la base de données. Le fichier se nomme « **database.php** » situé dans le dossier application en particulier dans le répertoire « **config** ». L'image ci-dessous nous montre les détails sur la configuration.

```

73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'ctexte',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97

```

Figure 25 : Le fichier de configuration pour la connexion vers la base de données

5.2.3. Implémentation de l'application

5.2.3.1. La classe « enseignant » : Modèle

Elle facilite l'accès et la récupération des informations liées à l'enseignant au niveau de la base de données (voir figure 26).

```

5 class EnsModel extends CI_Model {
6
7     // gestion des niveaux (Classes)
8
9
10    public function dataEnsNiv($numNiveau, $numEns) {
11        $this->db->select('
12            departement.numDept,
13            departement.nomDept,
14            filiere.numFiliere,
15            filiere.nomFiliere,
16            niveau.numNiveau,
17            niveau.numFiliere,
18            niveau.nomNiveau');
19        $this->db->from('matiere');
20        $this->db->join('niveau', 'niveau.numNiveau = matiere.numNiveau');
21        $this->db->join('filiere', 'filiere.numFiliere = niveau.numFiliere');
22        $this->db->join('departement', 'departement.numDept = filiere.numDept');
23        $this->db->where('matiere.numNiveau', $numNiveau);
24        $this->db->where('matiere.numEns', $numEns);
25        $query = $this->db->get();
26        return $query->result_array();
27    }
28
29    public function listNivDistinct($numEns) {
30        $this->db->select('matiere.numNiveau');
31        $this->db->distinct();
32        $this->db->from('enseignant');
33        $this->db->join('matiere', 'matiere.numEns = enseignant.numEns');
34        $this->db->join('niveau', 'niveau.numNiveau = matiere.numNiveau');
35        $this->db->join('filiere', 'filiere.numFiliere = niveau.numFiliere');
36        $this->db->join('departement', 'departement.numDept = filiere.numDept');
37        $this->db->where('matiere.numEns', $numEns);
38        $query = $this->db->get();
39        return $query->result_array();
40    }

```

Figure 26 : La classe « enseignant » : Modèle

5.2.3.2. La classe « enseignant » : Vue

C'est une page permettant de charger les données de l'enseignant tel que le prénom, le nom, la spécialité, la catégorie ainsi que le département auquel il est rattaché (voir figure 27).

```

128 <?php foreach ($listEns as $value) ( ?>
129     <tr>
130         <td>
131             <a onclick="return confirm('Etes-vous sûr de vouloir supprimer ?')"
132                 href="<?php echo base_url(); ?>index.php/ResController/delEns/<?php echo $value["numEns"]; ?>"
133                 class="btn btn-light btn-circle btn-sm">
134                 <i class="fas fa-trash"></i>
135             </a>
136         </td>
137         <td><?php echo $value["prenomEns"]; ?></td>
138         <td><?php echo $value["nomEns"]; ?></td>
139         <td><?php echo $value["specialite"]; ?></td>
140         <td><?php echo $value["categorie"]; ?></td>
141         <td><?php echo $value["nomDept"]; ?></td>
142     </td>
143     <td>
144         <a href="<?php echo base_url(); ?>index.php/ResController/editEns/<?php echo $value["numEns"]; ?>"
145             class="btn btn-light btn-circle btn-sm">
146             <i class="fas fa-edit"></i>
147         </a>
148     </td>
149 </tr>
</tr>
<?php } ?>

```

Figure 27 : La classe « enseignant » : Vue

5.2.3.3. La classe « enseignant » : Contrôleur

Cette page est dédiée au traitement des données de l'enseignant. Au finish, elles seront transmises à la vue qui gère l'affichage (voir figure 28).

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class EnsController extends CI_Controller {
6
7     public function __construct() {
8         parent::__construct();
9         $this->load->model('RespClasModel', 'RCMod');
10        $this->load->model('ResModel', 'RMod');
11        $this->load->model('EnsModel', 'ENSMoMod');
12        $this->load->helper('url');
13    }
14
15    public function index() {
16        $numEns = $this->session->userdata('numero');
17
18        $stablistNivDistinct = $this->ENSMoMod->listNivDistinct($numEns);
19
20        $i = 0;
21        $stabListNiv= array();
22        foreach ($stablistNivDistinct as $value) {
23            $stabListEnsNiv = $this->ENSMoMod->dataEnsNiv($value['numNiveau'], $numEns);
24
25            foreach ($stabListEnsNiv as $value) {
26
27                $stabListNiv['listNiv'][$i]['nomDept'] = $value['nomDept'];
28                $stabListNiv['listNiv'][$i]['numDept'] = $value['numDept'];
29                $stabListNiv['listNiv'][$i]['nomNiveau'] = $value['nomNiveau'];
30                $stabListNiv['listNiv'][$i]['numNiveau'] = $value['numNiveau'];
31                $stabListNiv['listNiv'][$i]['nomFiliere'] = $value['nomFiliere'];
32                $stabListNiv['listNiv'][$i]['numFiliere'] = $value['numFiliere'];
33            }
34            $i++;
35        }
36
37        $this->load->view('indexBens', $stabListNiv);

```

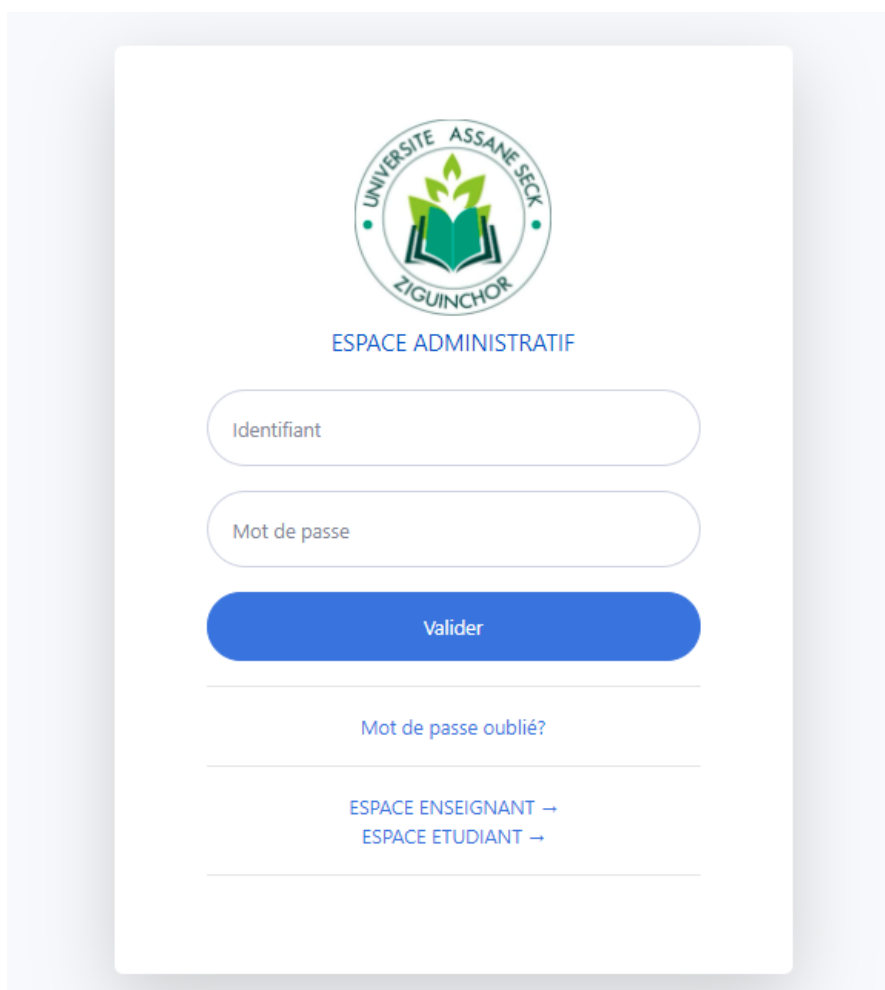
Figure 28 : La classe « enseignant » : Contrôleur

5.2.4. Présentation de quelques interfaces de l'application

5.2.4.1. Espace administratif

a. La page d'authentification

C'est une page qui renferme deux composantes essentielles : l'identifiant et le mot de passe. L'accès n'est autorisé qu'aux utilisateurs disposant d'un compte. Cependant, le compte des chefs de département, des responsables de formation, des responsables pédagogiques et des enseignants sont gérés par l'administrateur. Quant à celui des étudiants, il est impératif de passer par un formulaire d'inscription visible dans l'espace étudiant et ce compte est par la suite activé par le chef de département ou le responsable de formation (voir figure 29).



UNIVERSITE ASSANE SECK
ZIGUINCHOR

ESPACE ADMINISTRATIF

Identifiant

Mot de passe

Valider

Mot de passe oublié?

ESPACE ENSEIGNANT →
ESPACE ETUDIANT →

Figure 29 : La page d'authentification

b. La page d'accueil de l'administrateur

Comme son nom l'indique, il est chargé d'administrer tout le système. À partir du tableau de bord, il peut déterminer en terme d'effectif : le nombre d'enseignants (permanent ou vacataire), d'étudiants et de professionnels disponibles en temps réel. En dehors de cela, l'administrateur gère aussi (voir figure 30) :

- ♣ les utilisateurs (responsable de formation et pédagogique, chef de département) ;
- ♣ les UFR ;
- ♣ les départements ;
- ♣ les formations ;
- ♣ les enseignants.

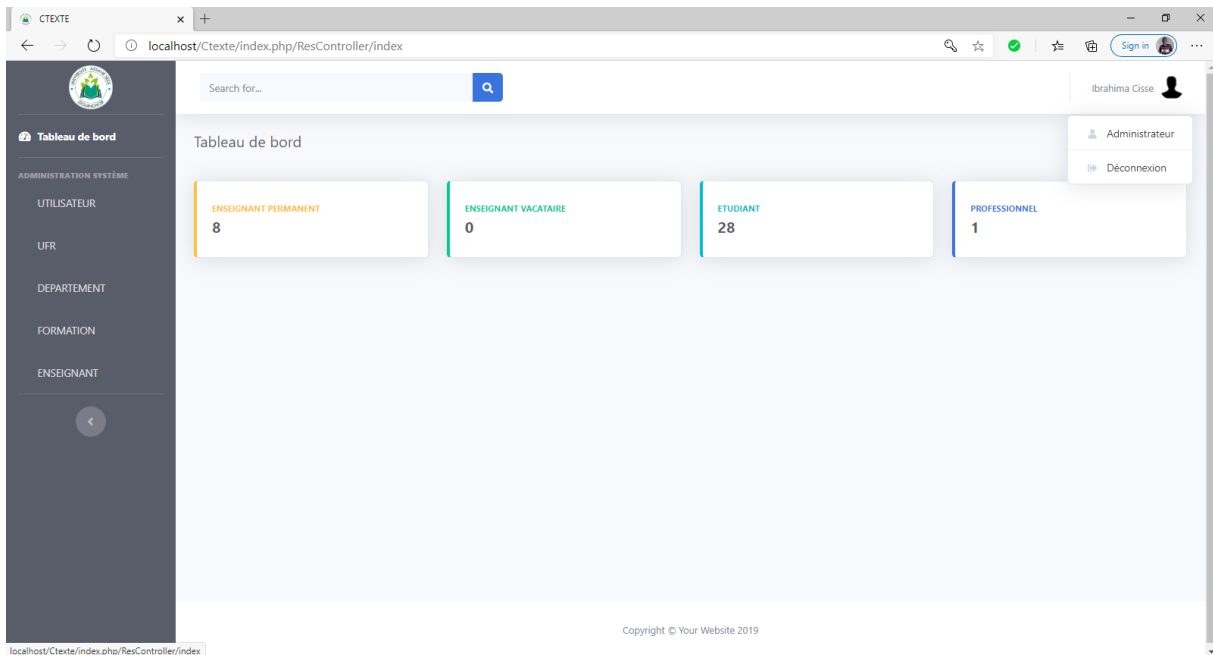


Figure 30 : La page d'accueil de l'administrateur

c. La page d'accueil du responsable de formation

C'est une page qui regroupe l'ensemble des formations qui sont sous la directive du responsable de la formation. En terme de fonctionnalité, elle est identique à celle du chef de département. Contrairement à la page d'accueil du responsable de formation, celle du chef de département recense toutes les formations du département. À partir de leur tableau de bord, ils peuvent (voir figure 31) :

- ♣ visualiser tous les enseignants de tous les départements ;
- ♣ affecter des matières aux enseignants ;
- ♣ consulter et évaluer les cours déjà fait par l'enseignant pour un niveau (classe) de formation ;
- ♣ visualiser les étudiants inscrits pour un niveau de formation ;
- ♣ activer le compte des étudiants.

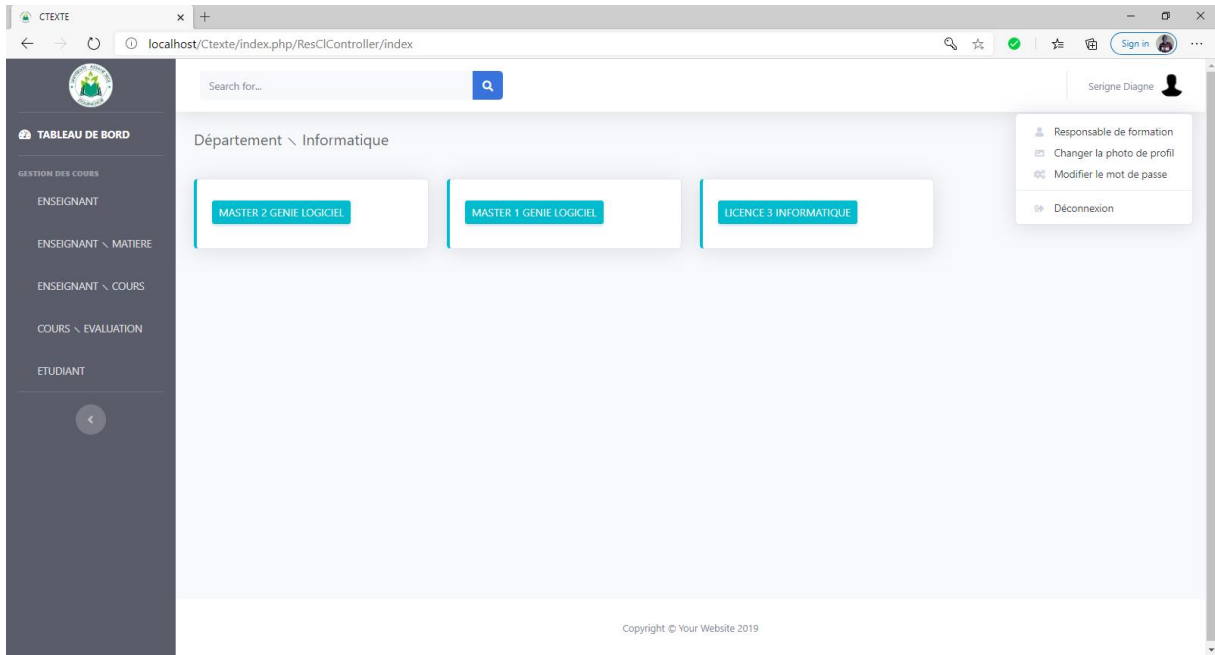


Figure 31 : La page d'accueil du responsable de formation

d. La page gestion enseignant et matière du responsable de formation

Elle énumère toutes les matières assurées par l'enseignant et attribuées par le responsable de formation pour un niveau de formation (voir figure 32).

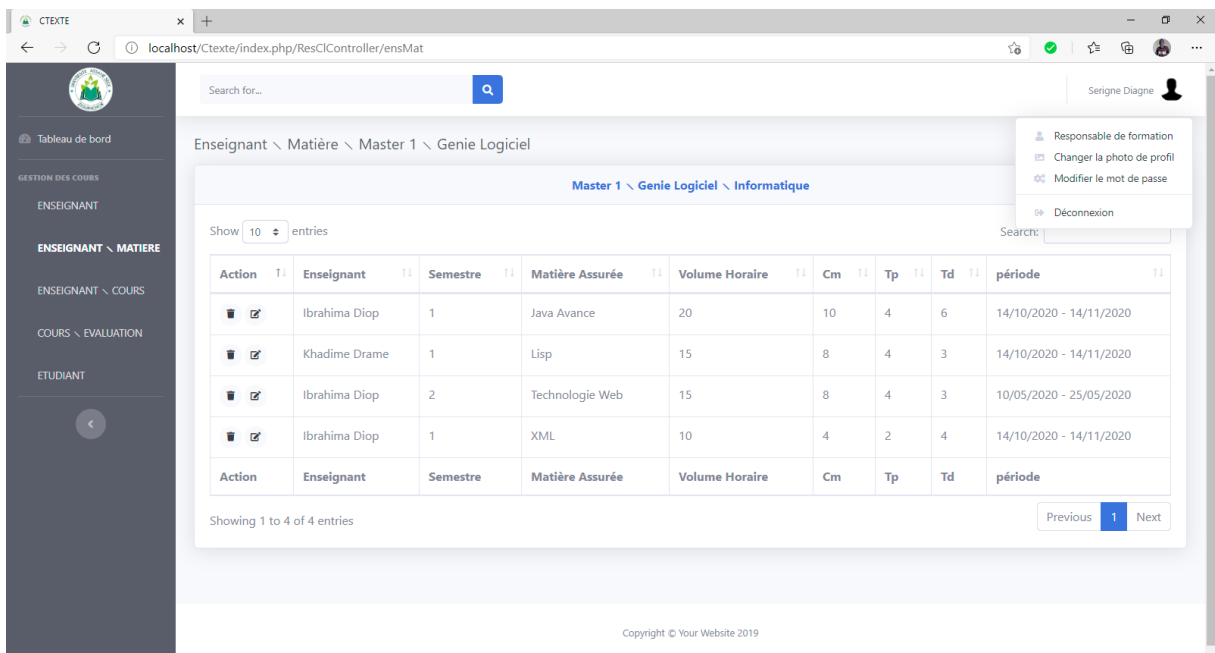


Figure 32 : La page gestion enseignant et matière du responsable de formation

e. La page d'accueil du responsable pédagogique

Elle recueille l'ensemble des départements dont la charge lui est confiée. Donc, à partir de cette page, il a le contrôle des départements en terme de gestion (voir figure 33) :

- ♣ des salles ;
- ♣ des niveaux ;
- ♣ des étudiants ;
- ♣ des nominations des responsables de classe.

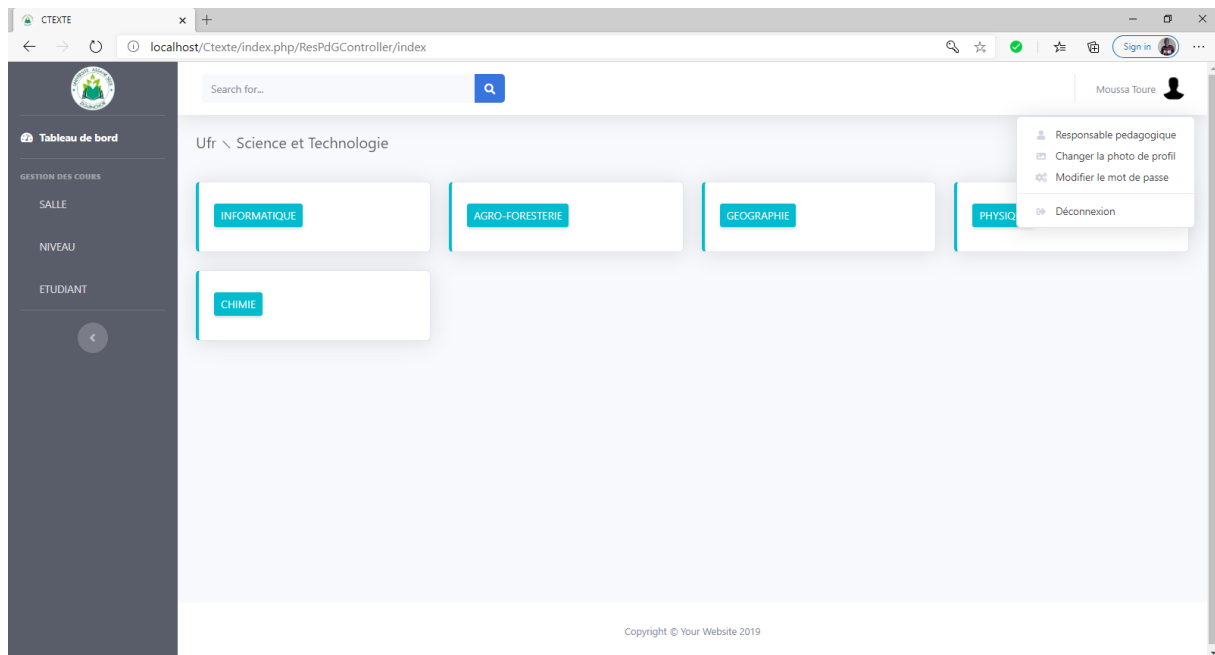


Figure 33 : La page d'accueil du responsable pédagogique

5.2.4.2. Espace étudiant

a. La page d'accueil de l'étudiant

C'est une page qui permet à l'étudiant d'avoir une vue globale de son espace de travail. À travers cette page l'étudiant peut (voir figure 34) :

- ♣ consulter et évaluer les cours déjà fait par l'enseignant (voir figure 35 et 37) ;
- ♣ télécharger les supports de cours (voir figure 35 ou 36) ;
- ♣ remplir le cahier de texte (droit réservé uniquement au responsable de la classe) (voir figure 36).

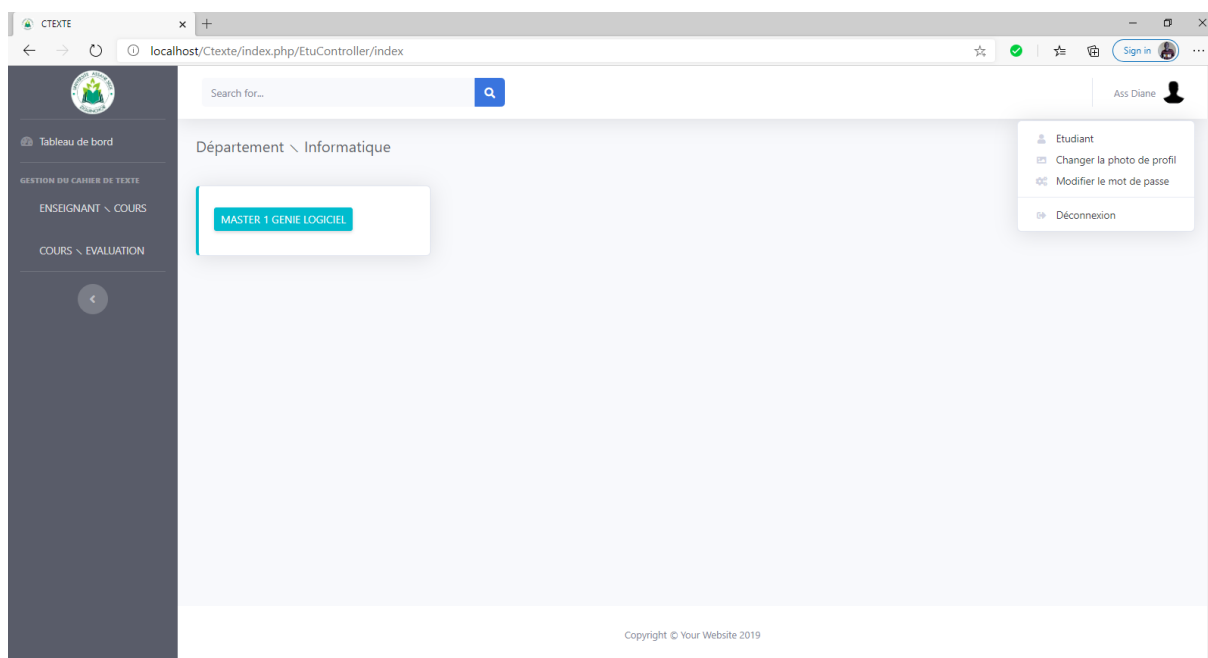


Figure 34 : La page d'accueil de l'étudiant

b. La page gestion enseignant et cours pour l'étudiant

Elle leur offre la possibilité de consulter les cours déjà faits par l'enseignant, mais aussi de télécharger les supports de cours s'ils sont disponibles (voir figure 35).

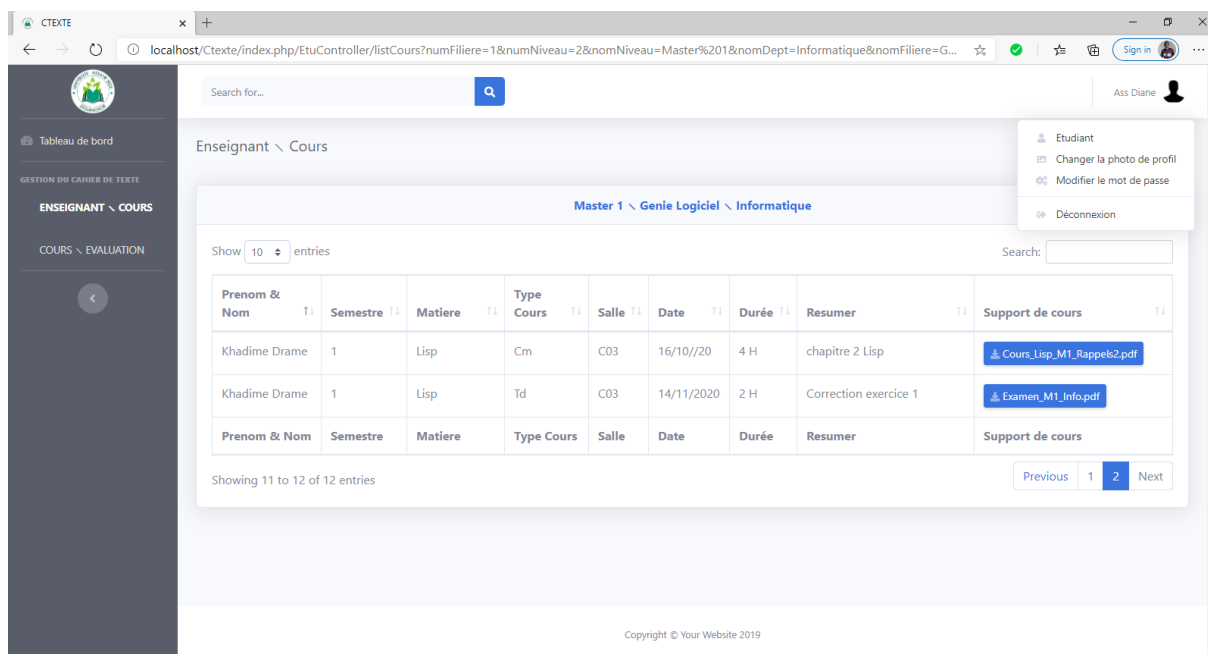


Figure 35 : La page gestion enseignant et cours pour l'étudiant

c. La page gestion enseignant et cours pour le responsable de la classe

Elle est presque identique à la page gestion « enseignant et cours » pour l'étudiant. La différence est que seul le responsable a le droit de remplir le cahier de texte, de modifier ou de supprimer le contenu (voir figure 36).

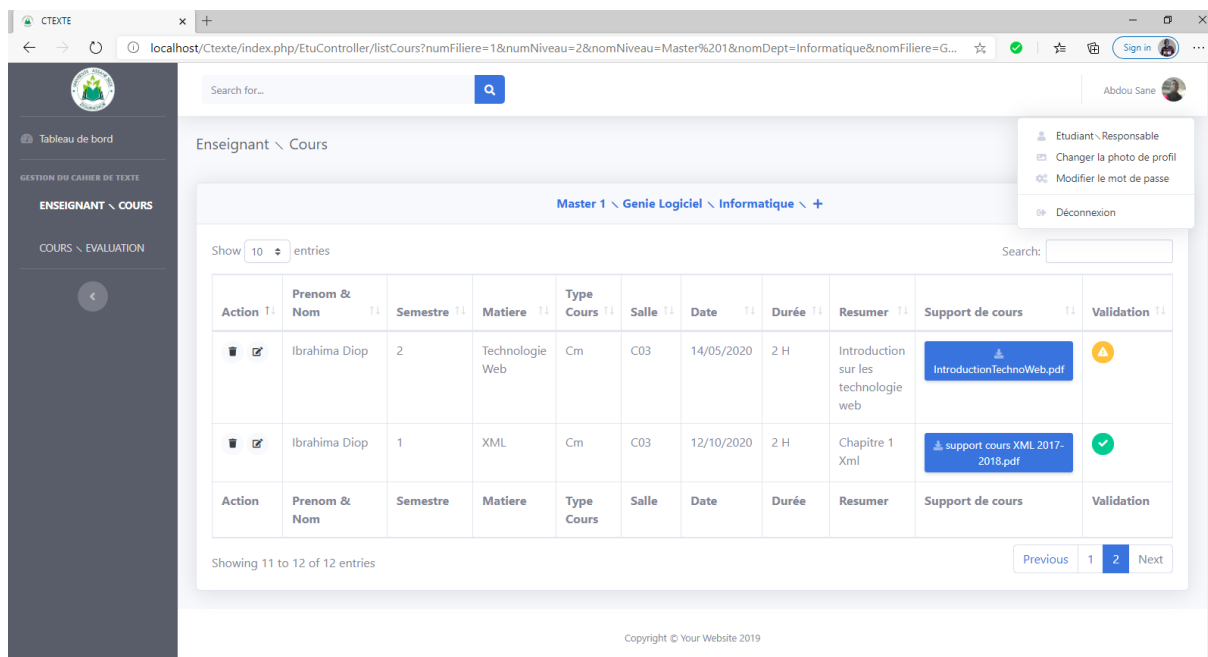


Figure 36 : La page gestion enseignant et cours pour le responsable de la classe

d. La page gestion cours et évaluation pour le responsable de la classe

Elle évalue les cours en termes de volume horaire fixé lors de la répartition des enseignements. En effet, le remplissage du cahier du texte électronique se fait en deux étapes. A la fin du cours, le responsable de la classe remplit le cahier de texte et ensuite, l'enseignant valide le contenu. Au fur et à mesure qu'il valide, l'application comptabilise ses heures de cours. Du coup, il arrivera un moment où le responsable de la classe ne pourra plus remplir, car ayant atteint le volume horaire qui a été fixé pour l'enseignant. Des codes couleurs sont mis en place pour renseigner sur une matière (voir figure 37) :

- ♣ Verte pour une matière en cours ;
- ♣ Orange pour une matière terminée ;
- ♣ Rouge pour une matière en retard de son exécution ou désactivée ;

Il faut noter également que cette page est identique à la page, gestion «cours et évaluation » du responsable de formation.

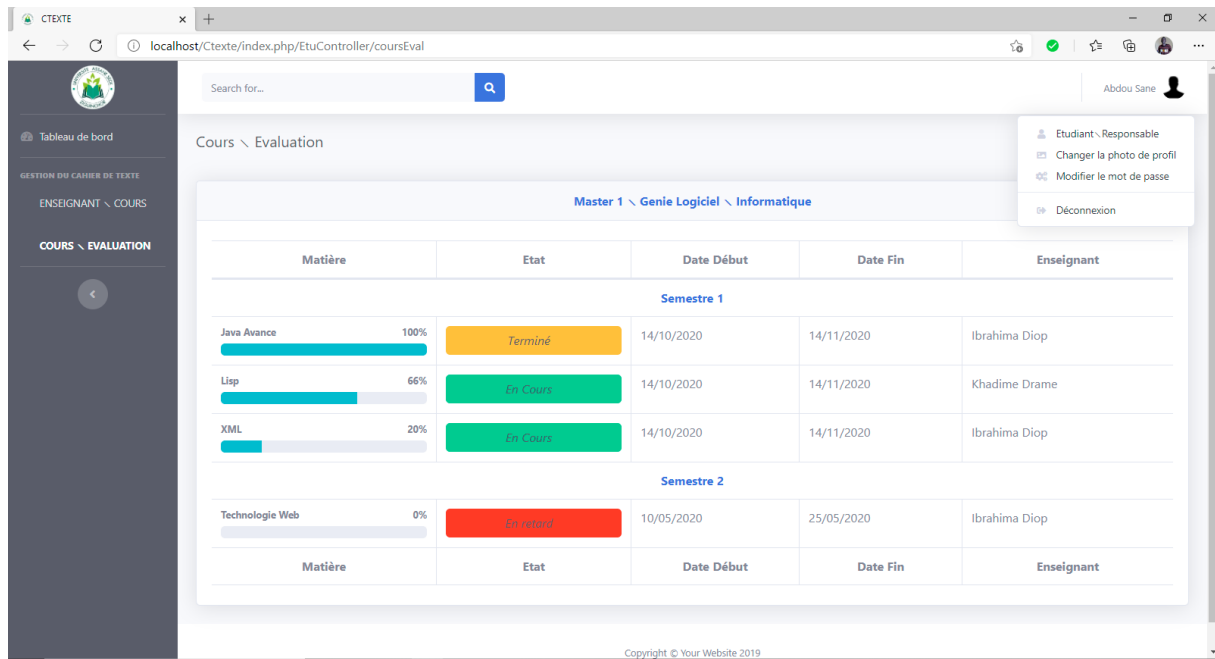


Figure 37 : La page gestion cours et évaluation pour le responsable de la classe

5.2.4.3. Espace enseignant

a. La page d'accueil enseignant

Elle permet à l'enseignant d'avoir le contrôle sur l'ensemble des niveaux de formation placé sous sa responsabilité (voir figure 38). Cependant, il a la possibilité :

- ♣ d'imprimer la fiche de déclaration des heures de cours avec l'option (voir figure 39) :
 - « **Fiche de déclaration S1** » pour imprimer tous les cours (matière, niveau, Nombre heure CM, Nombre heures TP, Nombre heures TD) dispensés par l'enseignant pour le semestre 1 ;
 - « **Fiche de déclaration S2** » pour imprimer tous les cours dispensés par l'enseignant pour le semestre 2 ;
- ♣ de consulter et d'évaluer les cours déjà fait (voir figure 40 et 41);
- ♣ de valider le contenu du cahier de texte (voir figure 40);
- ♣ de visualiser les étudiants de sa classe (voir figure 43);
- ♣ d'envoyer des mails aux étudiants (voir figure 45).

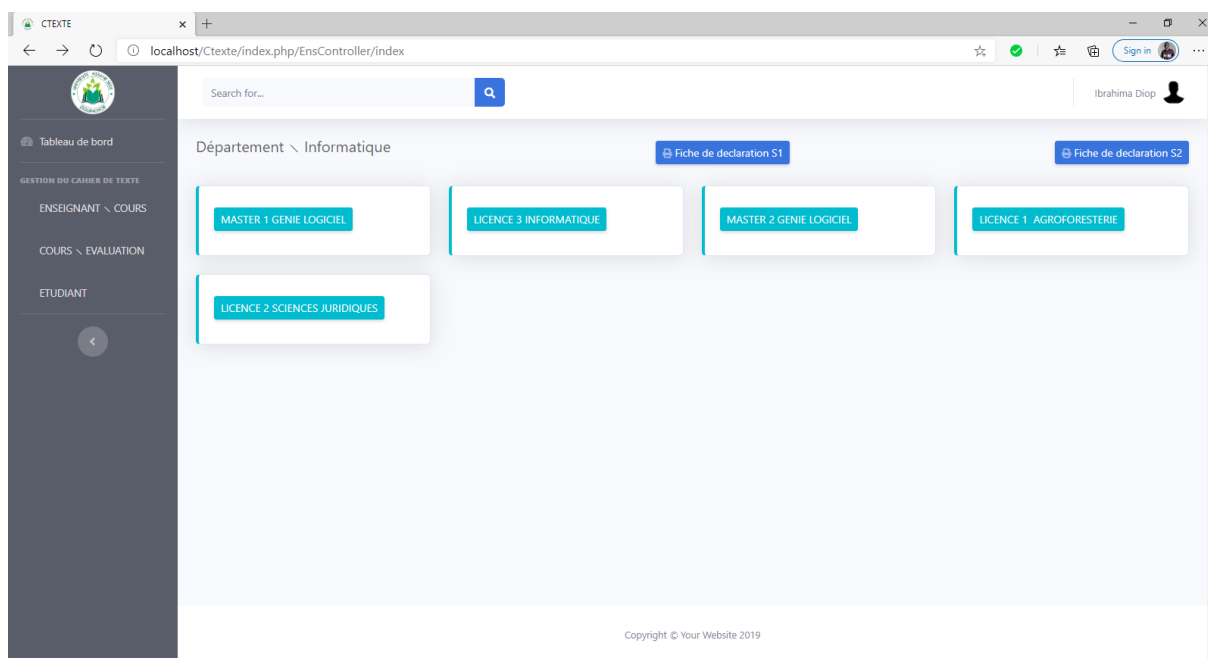


Figure 38 : La page d'accueil enseignant

b. La page impression : Fiche de déclaration

À travers cette fiche, l'enseignant aura un aperçu sur le nombre d'heures total effectué en CM, en TD et TP pour un semestre donné ainsi que les heures supplémentaires (voir figure 39).

Matière	Niveau	Nombre heures CM	Nombre heures TP	Nombre heures TD
Architecture Logiciel	Master 2 Genie Logiciel	30 H	0 H	0 H
Java Avance	Master 1 Genie Logiciel	10 H	4 H	6 H
XML	Master 1 Genie Logiciel	2 H	0 H	0 H
Php	Licence 3 Informatique	18 H	0 H	2 H
Informatique	Licence 2 Sciences Juridiques	6 H	4 H	0 H
Total général		66	8	8
Total en CM		76.4	-	-
Heures supplémentaires		4.4	-	3.52

Figure 39 : La page impression fiche de déclaration

c. La page gestion enseignant et cours : validation

C'est à partir de cette page, où l'enseignant pourra valider le contenu du cahier de texte déjà rempli par le responsable de la classe. Il est représenté par deux codes de couleur (voir figure 40):

- ♣ **Verte** : le cours est valide ;
- ♣ **Orange** : le cours est invalide.

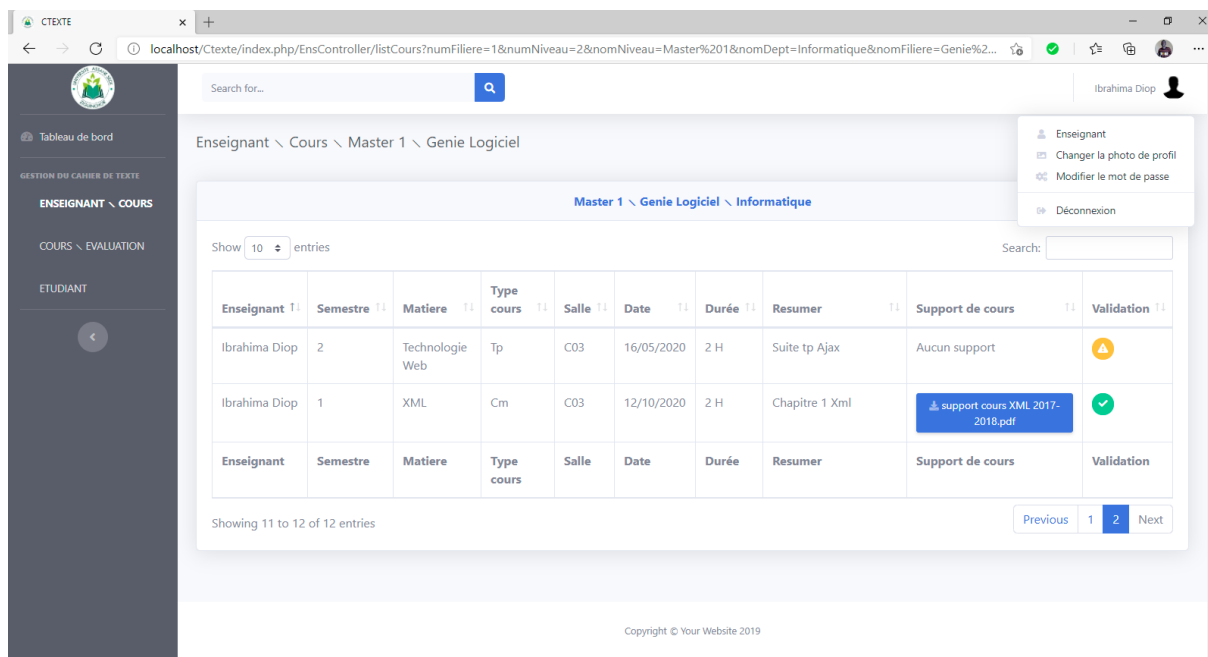


Figure 40 : La page validation

d. La page gestion cours et évaluation

Elle est presque identique à la page, gestion « cours et évaluation » du responsable de formation. La différence en est que l'enseignant a la possibilité d'imprimer la fiche de déclaration des heures de cours avec l'option (voir figure 41) :

- ♣ « **Fiche de déclaration S1 détaillée** » pour imprimer tous les cours (Matière, Niveau, Heure due, Heure fait (hr = heure restant), Période) dispensés par l'enseignant avec les détails (date, matière, type de cours, salle, durée) pour le semestre 1 d'un niveau de formation.
- ♣ « **Fiche de déclaration S2 détaillée** » pour imprimer tous les cours dispensés par l'enseignant avec les détails pour le semestre 2 d'un niveau de formation.

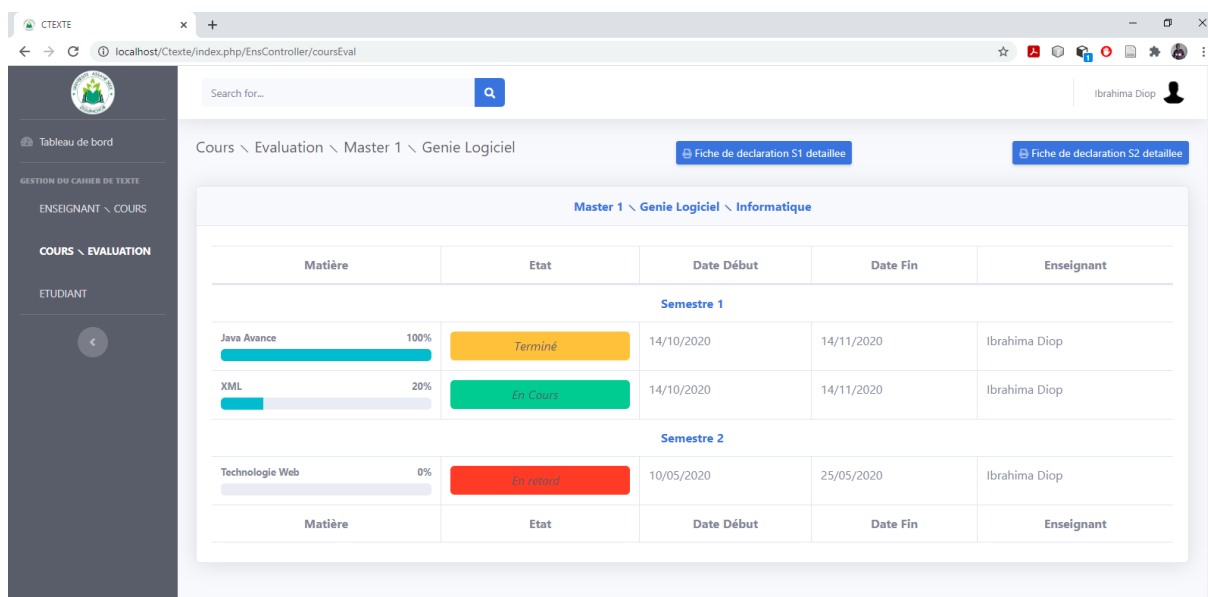


Figure 41 : La page gestion cours et évaluation

e. La page impression : Fiche de déclaration détaillée

Avec ce document, l'enseignant sait le nombre d'heures dues, le nombre d'heures fait et le nombre d'heures restant dans un niveau de formation et pour un semestre donné. Il aura également à sa disposition les détails (date, matière, type cours, salle durée) sur le déroulement des cours. Selon le semestre, ces documents (la fiche de déclaration et les fiches de déclaration détaillée) sont par la suite signés par le chef de département afin de permettre à l'enseignant de pouvoir percevoir son dû (voir figure 42).

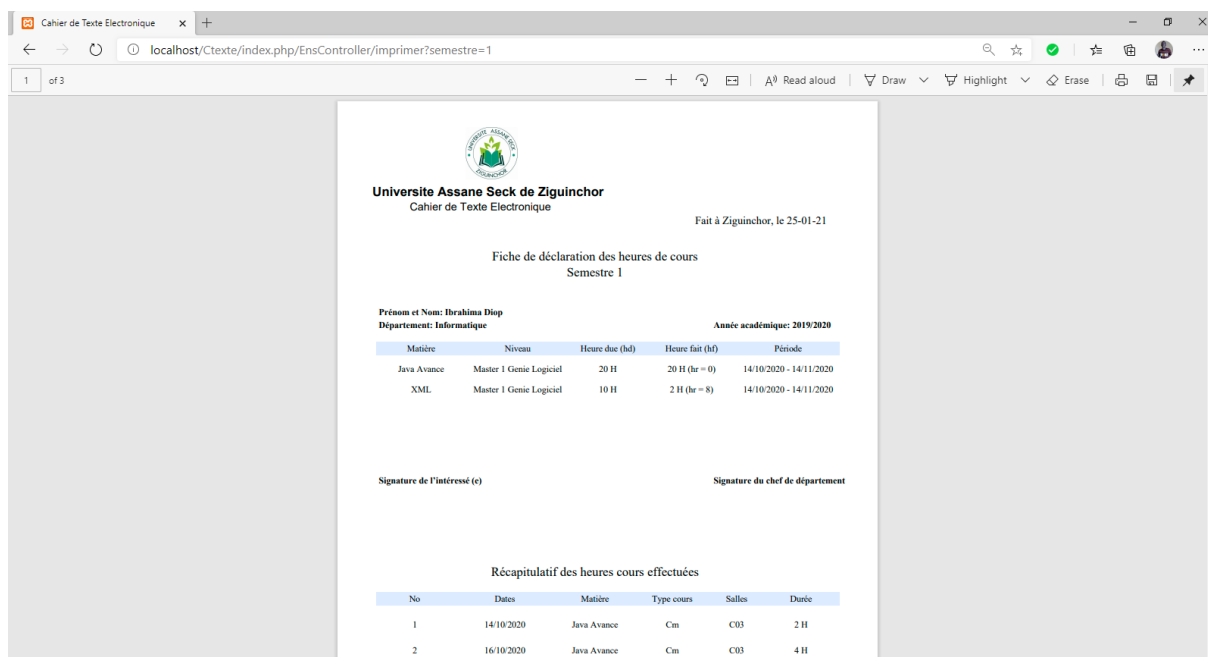


Figure 42 : La page impression fiche de déclaration détaillée

f. La page étudiant

C'est une page qui permet à l'enseignant d'avoir la liste des étudiants inscrits pour un niveau de formation (voir figure 43). Elle lui permet également de pouvoir exporter (grâce à l'onglet « exporter ») les données des étudiants (code étudiant, prénom, nom) sous format Excel à toutes fins utiles (voir figure 44).

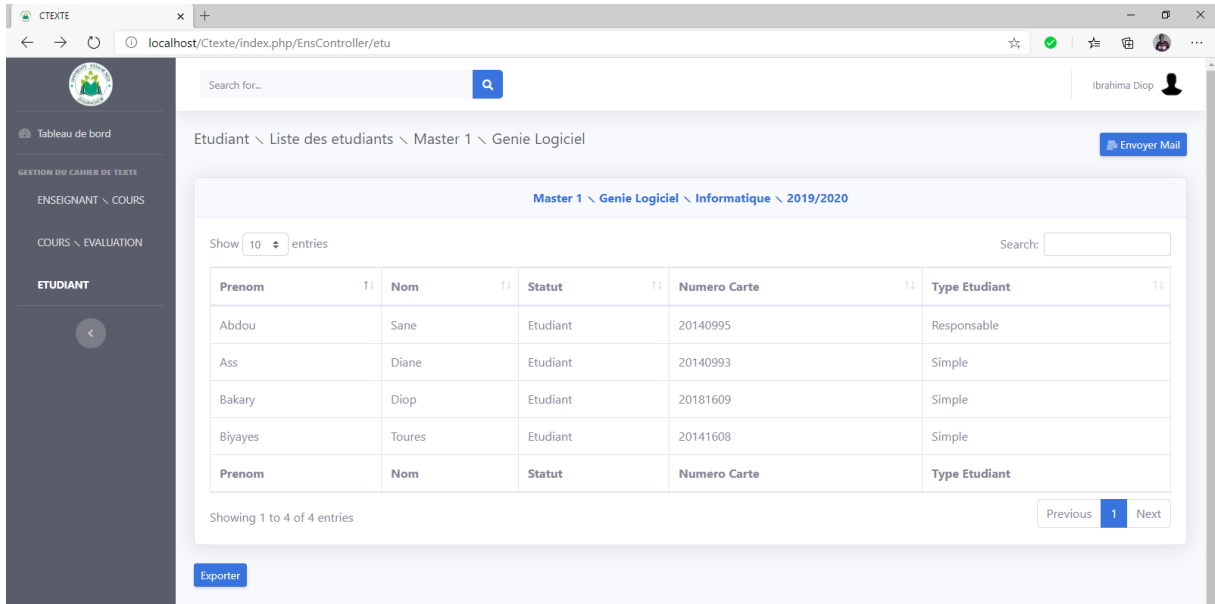


Figure 43 : La page étudiant

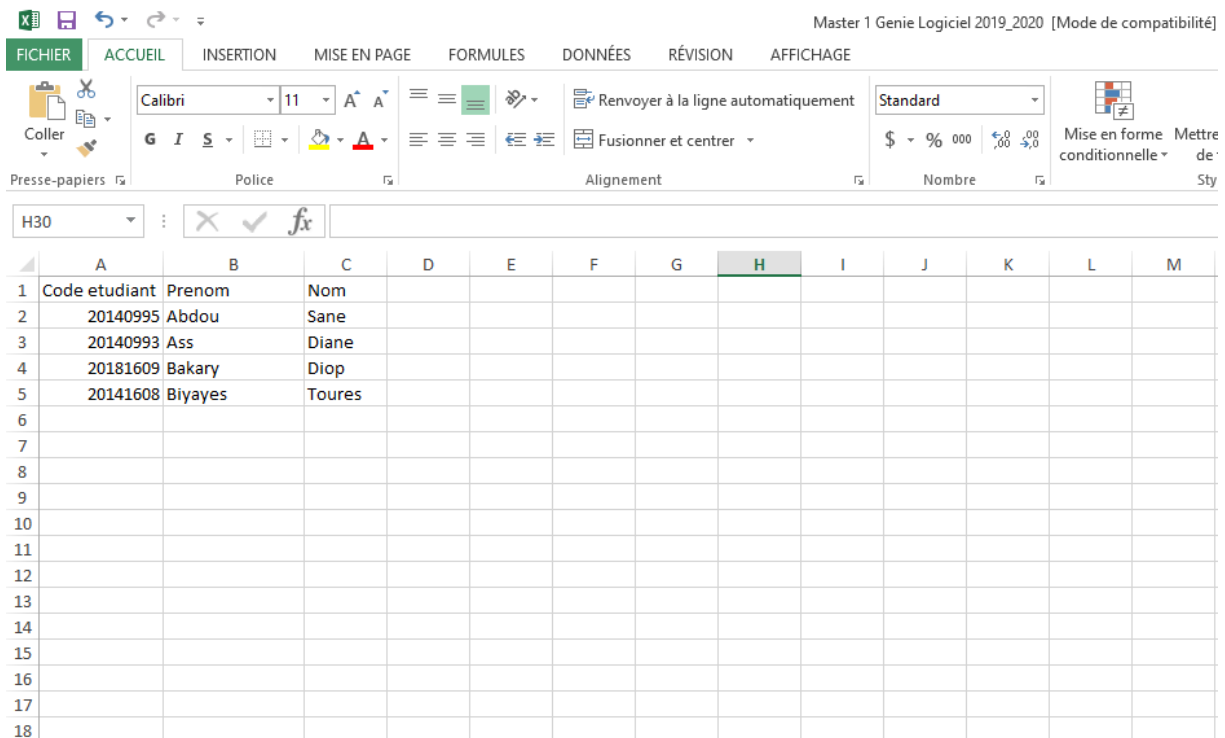


Figure 44 : Capture d'écran du contenu du fichier Excel

g. La page envoyer mail

À partir de la page étudiant (voir figure 43), tout à fait à droite, l'enseignant dispose d'un onglet « envoyer mail » pour informer ses étudiants (voir figure 45).

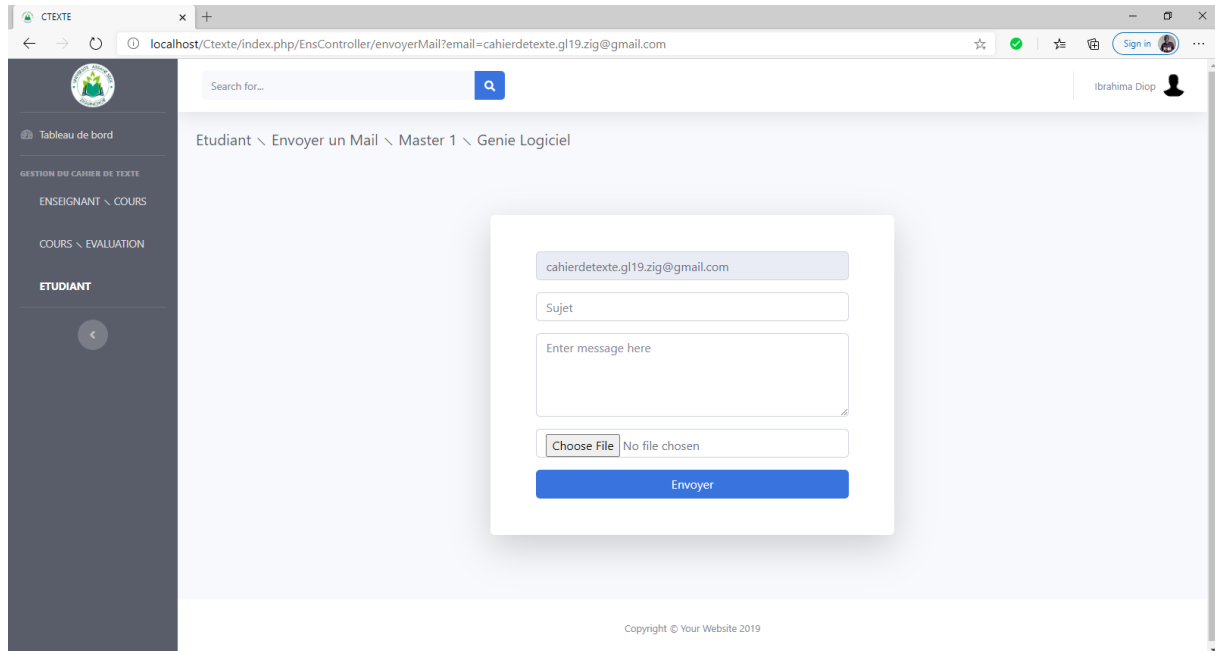


Figure 45 : La page envoyer mail

5.3. Conclusion

Dans ce chapitre, nous avons montré de manière détaillée les étapes qui nous ont permis de réaliser notre application. Nous avons commencé par lister les technologies utilisées avant de montrer l'installation et la configuration du Framework CodeIgniter. Nous avons ensuite présenté le processus de l'implémentation de la base de données puis de l'application. Enfin, nous avons présenté quelques interfaces de l'application. Le dernier chapitre de ce document est une conclusion générale résumant l'ensemble du travail accompli et des perspectives envisagées dans l'avenir.

Conclusion générale et perspectives

La réalisation de ce mémoire a duré plus que nous l'avions prévu. En réalité, il nous a fallu trois mois pour analyser le problème auquel s'attache notre travail de conception du futur système.

Ce mémoire nous a permis de mettre en place une application de gestion et suivi des cours, autrement dit la mise en place d'un cahier de texte électronique. Cette application permet aux étudiants, de pouvoir disposer d'un compte, de pouvoir y accéder, de consulter les cours et leurs états d'avancement, mais aussi de télécharger les supports de cours sans difficulté. Cette application permettra également aux personnels administratifs et aux personnels enseignants de travailler dans de meilleures conditions.

Grâce à ce travail, nous avons pu davantage nous familiariser au langage PHP et au Framework CodeIgniter. Néanmoins, il est important de savoir qu'une application n'est jamais finie même si nous avons géré toute l'expression des besoins qui ont été définis dans le cahier de charge en annexe B de ce document. Une application évolue dans le temps et d'autres contraintes, de nouvelles fonctionnalités vont apparaître.

En perspective, nous envisageons d'améliorer dans le futur l'application :

- ♣ **en renforçant le système de sécurité.** Car d'après l'ANSSI le nombre d'interventions menées en 2019 comporte 2296 signalements, 370 incidents hors opérateurs d'importance vitale, 9 incidents majeurs et 16 opérations de cyberdéfense.
- ♣ **en élargissant le domaine d'application.** Ceci étant dit, faire de telle sorte que les autres universités puissent adopter le même système d'information.
- ♣ **en mettant en place la version mobile de l'application.** Cependant, elle ne pourra pas prendre en charge toutes les fonctionnalités de la version web, mais quelques tâches seront opérationnelles.

Annexes

Annexe A : Dictionnaire de données

Nom	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			E	C			
Numéro		254					Utilisateur
Nom	Alphabétique	//	X				//
Prénom	Alphabétique	//	X				//
Identifiant	Alphanumérique	//	X				//
MotDePasse	Alphanumérique	//	X				//
Photo	Alphanumérique	//	X				//
Profil	Alphabétique	//	X				//
numFiliere	Numérique		X				Formation
nomFiliere	Alphabétique	//	X				//
numUfr	Numérique		X				Ufr
nomUfr	Alphabétique	//	X				//
numDept	Numérique		X				Département
nomDept	Alphabétique		X				//
numNiveau	Numérique		X				Niveau
nomNiveau	Alphanumérique		X				//
mailGroup	Alphanumérique		X				//
annee	Numérique		X				//
numMatiere	Numérique		X				Matière
nomMatiere	Alphabétique	//	X				//
timing	Numérique		X				//
cm	Numérique		X				
tp	Numérique		X				
td	Numérique		X				
debutMatiere	Alphanumérique	//	X				//
finMatiere	Alphanumérique	//	X				//
numEns	Numérique		X				Enseignant
nomEns	Alphabétique	//	X				//
prenomEns	Alphabétique	//	X				//
civilite	Alphabétique	//	X				//
specialite	Alphabétique	//	X				//
categorie	Alphabétique	//	X				//
identifiant	Alphanumérique	//	X				//
motDePasse	Alphanumérique	//	X				//
courriel	Alphanumérique	//	X				//
photo	Alphanumérique	//	X				//
numSalle	Numérique		X				Salle
nomSalle	Alphanumérique	//	X				//
capacite	Numérique		X				//
lieu	Alphabétique		X				//
numCour	Numérique		X				Cours

typeCours	Alphabétique		X				
dates	Alphanumérique	//	X				//
duree			X				//
resume	Numérique		X				//
validation	Alphabétique	//	X				//
numEtu	Numérique		X				Etudiant
prenomEtu	Alphabétique	//	X				//
nomEtu	Alphabétique	//	X				//
typeEtu	Alphabétique	//	X				//
statut	Alphabétique	//	X				//
identifiant	Alphanumérique	//	X				//
motDePasse	Alphanumérique	//	X				//
numeroCarte	Numérique		X				//
etat	Alphabétique	//	X				//
photo	Alphabétique	//	X				//

NB : E = Élémentaire et C = Calculée

Annexe B : Cahier de charge

SUJET : GESTION ET SUIVI DES COURS

On veut automatiser la gestion des cours au niveau de l'UFR Sciences et Technologies de l'Université Assane Seck de Ziguinchor, c'est-à-dire la création d'un cahier de texte électronique. L'application permettra de suivre les cours dispensés et de voir leurs états d'avancement en fonction du timing fixé. En outre, elle permettra l'ajout des supports de cours sous format électronique de chaque département, de chaque filière à tous les niveaux (L1, L2, L3, M1 et M2) que l'étudiant pourra récupérer en se connectant (login et mot de passe valide) au niveau de l'application.

Chaque enseignant-chercheur possède un login et un mot de passe pour la gestion de ses propres cours. Le responsable de formation de chaque niveau et le responsable pédagogique sont identifiés eux aussi par un login et mot de passe unique, ils peuvent voir l'état d'avancement de chaque cours et de chaque classe.

Un formulaire d'inscription sera créé pour les étudiants, mais cette inscription sera validée par le responsable de niveau.

L'un des objectifs de l'application est de créer un tableau de bord pour l'ensemble des niveaux de chaque département, de chaque filière de l'UFR ST.

Des codes couleurs pourront renseigner sur un cours (module) par exemple :

- ♣ **Verte** pour un cours ou module ou matière en cours
- ♣ **Orange** pour une matière terminée
- ♣ **Bleu** pour un cours désactivé
- ♣ **Rouge** pour une matière en retard de son exécution

Le tableau de bord donne des renseignements sur chaque niveau, sur le responsable de niveau et sur les enseignant-chercheurs intervenants dans chaque classe.









Licence1 1 ère ANNEE INFORMATIQUE					
Pourcentage	Matières	Etat	Date Début	Date Fin	Enseignant-chercheur
	Développement web		14/01/2019	14/04/19	Mr DIOP
	Bases de Données		01/01/2019	4/04/2019	M. DIENG
	Langage C		3/02/2019	29/03/2019	M. DIATTA
	Mérise		4/010/2019	05/05/2019	M. GAYE

Figure 46 : Tableau de bord

REGLES DE GESTION

- ♣ Un niveau peut suivre une ou plusieurs matières.
- ♣ Un enseignant-chercheur peut dispenser une ou plusieurs matières dans une ou plusieurs classes.
- ♣ Une matière est assurée par un seul enseignant-chercheur dans chaque niveau (classe).
- ♣ Une matière a au moins un chapitre
- ♣ Un chapitre est constitué de plusieurs parties
- ♣ Une classe a un seul responsable.
- ♣ Un étudiant appartient à une seule classe.

Remarque : le cahier de charge a subi d'importante amélioration lors du processus de développement.

Bibliographie

- [1] M. Nebra, Réalisez votre site web avec HTML5 et CSS3, Edition OpenClassrooms, Collection Livre du Zero, Décembre 2011.
- [2] P. Roques, UML2 par la pratique, Editeur Eyrolles, Collection Noire, septembre 2009.
- [3] Thuillier, Victor, Programmez en orienté objet en PHP, Edition eyrolles, Collection OpenClassrooms, Mai 2017.
- [4] G. Benefield, C. Larman, and B. Vodde, Guide Léger de la Théorie et de la Pratique de Scrum, p. 22.
- [5] D. Upton, CodeIgniter 1.7. 2009

Webographie

- [6] <http://www.wikipedia.fr>, visité le 8/12/2020
- [7] <https://www.openclassrooms.com>, visité le 8/12/2020
- [8] <http://getbootstrap.com>, visité le 8/12/2020
- [9] https://codeigniter.com/userguide2/database/active_record.html, visité le 25/11/2020
- [10] <https://www.apachefriends.org/fr/index.html>, visité le 21/12/2020
- [11] <http://www.next-decision.fr/les-editeurs/autres/sap-power-amc>, visité le 8/12/2020
- [12] <http://www.jmrenouard.fr/wp-content/uploads/2011/05/codeigniter.pdf>, visité le 25/11/2020
- [13] <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html>, visité le 8/12/2020
- [14] <http://php.net/manual/fr>, visité le 8/12/2020
- [15] <https://netbeans.org>, visité le 8/12/2020
- [16] <https://www.nutcache.com/fr/blog/methodologie-scrum>, visité le 8/12/2020
- [17] <https://pyxis-tech.com/fr/approches-agiles>, visité le 8/12/2020
- [18] <https://www.univ-zig.sn>, visité le 8/12/2020
- [19] https://www.memoireonline.com/07/10/3700/m_Conception-et-realisation-dune-application-web-pour-la-gestion-des-stocks-cas-detude-magasin23.html, visité le 8/12/2020
- [20] <https://tcpdf.org>, visité le 28/11/2020
- [21] https://fr.wikibooks.org/wiki/Programmation_PHP/PHPExcel, visité le 24/1/2021
- [22] <https://www.ssi.gouv.fr>, visité le 27/1/2021