



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique



UNIVERSITÉ ECHAHID HAMMA LAKHDAR
ELOUED

FACULTÉ DES SCIENCES EXACTES

Domaine: Mathématiques et Informatique

Filière: Informatique

Spécialité: systèmes informatiques

Projet de fin d'étude

LICENCE ACADEMIQUE

Thème

**Développement d'une application Web de gestion des dossiers
administratif et pédagogique des étudiants**

Présenté par:

- ✓ Lamoudi Walid
- ✓ Arigue Mona
- ✓ Sahraoui Said

Encadreur:

Mr. Medileh Saci

**Soutenu le 23 mai 2016
Devant le jury composé de:**

KHEBBACHE Mohib Eddine	M.A.B	Président.
YAKOUB Mohammed amine	M.A.A	Examinateur

Année Universitaire: 2015 / 2016

Remerciements

Nous

Remercions tout d'abord notre dieu qui nous donné La force et la volonté pour élaborer ce travail.

Nous adressons nos vifs remerciements à notre encadreur

Mr Medilek Saci

qui nous a aidé toute la durée de notre travail et par patience et les précieux conseils dont il nous a entouré.

Notre reconnaissance va aussi à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'informatique université echahid hamma laḳdar.

On remercie également tous ceux qui ont participé de près ou de loin à élaborer ce travail.

Dédicace

Nous voulons présenter dans un premier temps nos dédicaces à notre encadreur « MR MEDJLEH Saci »

Nous voulons également lui témoigner notre gratitude pour sa patience et son soutien qui nous ont été précieux afin de mener notre travail à bon port , de tous nos respects aussi à :

Nos parents pour toute l'affection, tous les sacrifices et les encouragements qu'ils n'ont pas cessé de nous prodiguer tout au long de nos études

A tous ceux qui nous ont soutenues tout au long de ce projet.

Lamoudi Walid & Sakraoui Said

&

Arigue Mona

Tableau des figures

Nº	Figure	Page
01	Diagramme de cas d'utilisation " Gestion compte d'utilisateur " .	09
02	Diagramme de cas d'utilisation " Gestion de dossier d'un étudiant " .	10
03	Diagramme de cas d'utilisation " Consultation de dossier " .	11
04	Le diagramme de cas d'utilisation.	14
05	Le diagramme de classe.	19
06	Diagramme de cas d'utilisation " Connexion au site " .	22
07	Diagramme de séquence " Connexion au site " .	22
08	Diagramme de cas d'utilisation " Gestion compte d'utilisateur " .	23
09	Diagramme de séquence " Gestion compte d'utilisateur " .	23
10	Diagramme de cas d'utilisation " Gestion de dossier d'un étudiant " .	24
11	Diagramme de séquence " Inscription d'un étudiant " .	24
12	Diagramme de séquence " Mise à jour d'un dossier d'étudiant " .	25
13	Diagramme de cas d'utilisation " Consultation de dossier " .	26
14	Diagramme de séquence " Consultation de dossier " .	26
15	Interface de microsoft visual studio.	32
16	Interface de " Page d'accueil " .	37
17	Interface de " Log in " .	38
18	Interface de " Creation d'un compte" .	39
19	Interface de " Inscription d'un étudiant" .	40
20	Interface de " Update fiche " .	41

Liste des tableaux

Nº	Tableau	Page
01	Liste des cas d'utilisations.	13
02	Dictionnaire des données.	17
03	Table des classes candidates.	18
04	La base des données de l'application.	20

Sommaire

Introduction Générale	01
Chapitre I : Etude théorique et choix techniques	
1. Introduction	03
2. Etude Préliminaire	03
2.1 Etude de l'existant : Inscription des étudiants et traitement des dossiers	03
2.2 Critique de l'existant et solutions proposées	06
3. Conclusion	06
Chapitre II : Analyse et spécification des besoins	
1. Introduction	08
2. Analyse globale de l'application	08
2.1 Description des acteurs	08
2.2 Spécification des besoins fonctionnels	12
2.3 Spécification des besoins non fonctionnels	12
2.4 Diagramme des cas d'utilisation	13
Chapitre III : Conception	
1. Introduction	16
2. Conception détaillée	16
2.1 Dictionnaire des données	16
2.2 Le diagramme de classe	17
2.3 La base des données de l'application	20
2.4 Les diagrammes de séquence	21
3. Conclusion	27
Chapitre IV : Réalisation	
1. Introduction	29
2. Environnement de travail	29
2.1 Environnement matériel	29
2.2 Environnement logiciel	29
2.2.1 Langage de programmation (IDE)	32
2.2.1.1 Présentation	32
2.2.1.2 Système de gestion de base de données utilisé	33
2.2.1.3 Serveur d'application	34
3. Interfaces de l'application	38
4. Conclusion	43
Conclusion générale	44
Bibliographie	44

Introduction générale

L'informatique qui est devenue dans les dernières années plus importante dans notre vie parce qu'elle est une science de traitement de données par des outils automatique .Elle est utilisée dans nos jours comme un outil de gestion et de comptabilité et aussi pour la meilleure organisation des entreprises.

Le développement des entreprises ne sera pas réalisé que par bien occuper l' élément humain et les activités manuelles comme l' interaction avec les documents qui avoir une grande importance dans tel organisme et pour tel personne car ils représentent une source de décision . Pour ces influences directes sur le travail de l' administration et l' organisme générale, et pour cela on a développé une application web centralisée pour répondre aux besoins administratifs de gestion des dossiers des étudiants au niveau de département d'informatique de notre université.

Donc, pour réaliser ce travail nous avons suit le plan suivant :

- ✓ Le premier chapitre concerne l'étude théorique et les choix techniques.
- ✓ Le deuxième chapitre est basé sur l'analyse et la spécification des besoins.
- ✓ Le troisième chapitre représente la conception de l'application.
- ✓ Le dernier chapitre représente la réalisation et l'implémentation du système aussi bien quelques interfaces de notre application.

Enfin, nous conclurons ce projet par une conclusion générale.



Chapitre I

Etude théorique et choix
techniques

1. Introduction

Dans ce chapitre, nous allons représenter l'étude théorique et les choix techniques. Nous commencerons par l'étude de l'existant (inscription des étudiants et traitement des dossiers), puis nous passerons à la critique de l'existant et les solutions proposées.

2. Etude préliminaire

2.1. Etude de l'existant:

Actuellement, l'administration de notre université utilise des systèmes manuels dans le traitement des dossiers des étudiants, la manipulation des dossiers est effectuée par l'utilisation des papiers et des dossiers qui sont organisées dans des dossiers ce qui demande beaucoup de temps et constitue une tâche administrative dure.

D'après l'administration de notre université (département d'informatique)

- **Le premier cas : "Inscription d'étudiant"**

Première année d'étude:

✓ **الملف البيداغوجي : على مستوى مصلحة التعليم و التقييم**

1. نسخة من شهادة البكالوريا.

2. صورتان شمسيان (02).

3. شهادة تسجيل (خاص بالبيداغوجي).

✓ **الملف النهائي: على مستوى مصلحة الانخراط المركزي**

1. شهادة البكالوريا الأصلية.

2. نسختان من شهادة البكالوريا (02).

3. حقوق التسجيل 200 دج.

4. 04 صور شمسية.

5. تصريح شرفي.

6. شهادة توجيه.

بالإضافة إلى الوثائق المدخلة السالفة الذكر، يضاف إلى الملف النهائي:

- 02 شهادة مدرسية (شهادة للطالب وشهادة تضاف إلى الملف).
- وثيقة تبرير غياب.

Deuxième année d'étude:

حالتان: إعادة السنة أو النجاح.

✓ الملف البيداغوجي: على مستوى مصلحة التعليم والتقييم.

1. تبرئة مكتبة.
2. حقوق التسجيل 200 دج.
3. 02 صور شمسية.
4. نسخة من كشف النقاط.
5. شهادة إعادة التسجيل.
6. شهادة مدرسية.

وثائق أخرى يمكن إضافتها إلى الملف:

- وثيقة تبرير غياب.
- مقرر عطلة أكاديمية.
- مقرر استئناف عطلة أكاديمية.
- مقرر تأديبي.
- مقرر استئناف تأديبي.
- مقرر إعادة ادماج.

الوثائق المستخرجة:

- شهادة إعادة التسجيل.
- شهادة مدرسية.

Troisième année d'étude:

même procédure de deuxième année et même dossier.

- **Le deuxième cas: "Mise à jour des dossiers d'étudiants"**
- **Modification**

يمكن التغيير في:

- رقم التسجيل.
- اسم ولقب الطالب باللغتين العربية والفرنسية.
- اسم الاب.
- اسم و لقب الام.
- تاريخ و مكان الميلاد.
- الجنس.
- العنوان.
- شعبة البكالوريا.
- رقم البكالوريا.
- الولاية.
- علامة البكالوريا.
- سنة البكالوريا.

2.2. Critique de l'existant et solutions proposées:

D'après l'étude de l'existant, il est clair qu'il y a beaucoup d'inconvénients dans la méthode classique d'organisation des documents administratifs, qu' elle provoque les problèmes suivants:

- Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents .
- Recherche difficile sur les registres qui engendre une perte de temps.
- Un grand nombre des formulaires et documents qui engendrent une difficulté de stockage.
- La présence de plus d' une copie d' un seul document.
- Le retour à l'archive pour vérifier le dossier.

Notre principal objectif est de gérer automatiquement les solutions de ces problèmes, nous proposons donc un outil de gestion des documents qui sera en vigueur dans toute les unités administratives des établissements sera simple, uniforme et devra rencontrer les objectifs suivants:

- Traitement automatique des dossiers d' étudiants.
- Facilité l'accès aux documents et la recherche.
- Eviter la répétition du travail.

3. Conclusion

Dans le but de faciliter la gestion des dossiers des étudiants, dans ce chapitre ,nous avons critiqués l'existant et proposer des solutions pour la gestion automatique des dossiers en suivant une démarche de développement.



Chapitre II

Analyse et spécification des besoins

1. Introduction

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel.

Dans ce chapitre, nous abordons la phase d'analyse et spécification des besoins. Ainsi, nous présentons les besoins fonctionnels et non fonctionnels de notre application. Nous utilisons le langage UML comme un moyen simple et compréhensible afin de décrire les principaux cas d'utilisation .

2. Analyse globale de l'application

2.1. Description des acteurs:

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système ,mais d'abord nous donnons une définition de ce qu'un acteur .

Définition: Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou Autre système) qui interagissent directement avec le système étudié.[6]

Les acteurs de notre système sont:

- **Administrateur (Chef département):**

1- Gestion des comptes d'utilisateurs:

- ✓ Création de compte d'utilisateur.
- ✓ Modification de compte d'utilisateur.
- ✓ Suppression de compte d'utilisateur.

2- Gestion dossiers d'étudiants:

- ✓ Inscription d'étudiant.
- ✓ La mise à jour des dossiers d'étudiants.

3- Consultation des dossiers d'étudiants:

Après la consultation, l'administrateur peut imprimer le fichier qu'il veut.

- **Utilisateur (Secretariat):**

- ✓ Consultation des dossiers.

Après la consultation, l'utilisateur peut imprimer le fichier qu'il veut.

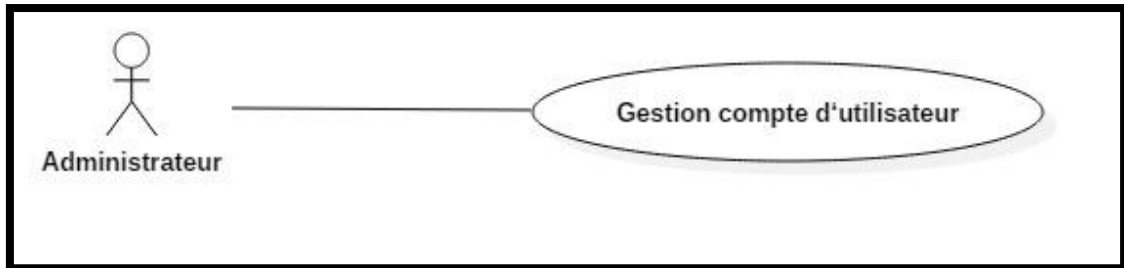
Cas d'utilisation 1 : " Gestion de compte d'utilisateur "

Figure 1: Diagramme de cas d'utilisation " Gestion compte d'utilisateur ".

- Objectif:

Permettre à l'administrateur de l'application de créer ,de modifier un compte d'utilisateur.

- Acteur concerné:

L'administrateur de l'application.

- Pré condition:

L'administrateur de l'application s'est authentifiée correctement à l'application.

- Scénario nominal:

Demander la page de gestion d'un compte d'utilisateur.

1- Le système affiche le formulaire de création/modification d'un compte d'utilisateur.

2- L'administrateur de l'application saisit les informations (création de compte) nécessaires et valide sa saisie.

3- Le système vérifie la présence des données obligatoires .

4- le système enregistre la saisie validée.

- Scénario alternatif:

2.a- Modification de compte d'utilisateur.

✓ Le système affiche le formulaire de saisie des données (création de compte) enregistrées.

✓ L'administrateur de l'application modifie les données.

✓ Le cas d'utilisation reprend à l'action 3 du scénario nominal.

3.a- Erreurs détecté dans la saisie.

- ✓ Le système réaffiche le formulaire de saisie en indiquant les erreurs détectées.
- ✓ L'administrateur de l'application corrige les erreurs détectées.
- ✓ Le cas d'utilisation reprend au point 3 du scénario nominal.

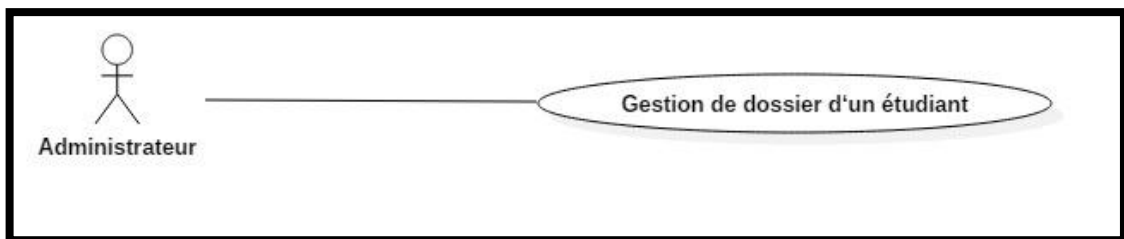
Cas d'utilisation 2: " Gestion des dossiers d'étudiants "

Figure 2: Diagramme de cas d'utilisation " Gestion de dossier d'un étudiant ".

- Objectif:

Permettre à l'administrateur de l'application d'inscrire les étudiants et faire l'aperçue et la mise à jour des dossiers des étudiants.

- Acteur concerné:

Administrateur de l'application.

- Pré condition:

L'administrateur de l'application s'est authentifiée correctement à l'application.

- Scénario nominal:

Demander la page d'inscription d'un étudiant.

- ✓ Le système affiche le formulaire de saisie des données(formulaire d'inscription étudiant).
- ✓ L'administrateur de l'application saisit les informations nécessaires et valide sa saisie.

- ✓ Le système vérifie la présence des données obligatoires.

- ✓ Le système enregistre la saisie validée.

- Scénario alternatif:

2.a- La mise à jour des dossiers d'étudiants (modification et suppression des dossiers).

- ✓ Le système affiche la liste des étudiants inscrits.

- ✓ L'administrateur de l'application choisit l'étudiant.
 - ✓ L'administrateur de l'application modifie/supprime le dossier concerné.
 - ✓ Fin de cas d'utilisation.
- 3.a- Erreurs détecté dans la saisie.**
- ✓ Le système réaffiche le formulaire de saisie en indiquant les erreurs détectées.
 - ✓ L'administrateur de l'application corrige les erreurs détectées.

Cas d'utilisation 3: " Consultation des dossiers d'étudiants "

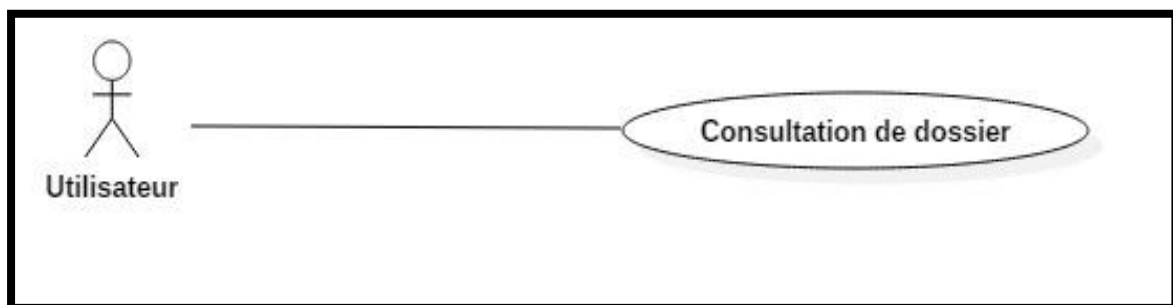


Figure 3: Diagramme de cas d'utilisation " Consultation de dossier ".

- Objectif:
Permettre à l'administrateur de l'application et à l'utilisateur de consulter les dossiers d'étudiants.
- Acteurs concernés:
 - ✓ Administrateur de l'application.
 - ✓ utilisateur de l'application.
- Pré condition:
 - ✓ L'administrateur de l'application s'est authentifiée correctement à l'application.
 - ✓ L'utilisateur de l'application s'est authentifiée correctement à l'application.
- Scénario:
 - ✓ L'utilisateur demande la consultation et choisit l'étudiant.
 - ✓ Le système affiche la liste des dossiers.
 - ✓ L'utilisateur choisit le dossier.
 - ✓ Le système affiche le dossier.
 - ✓ Fin de cas d'utilisation.

2.2. Spécification des besoins fonctionnels:

Elle s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

- Le système doit permettre:
 - ⇒ A l'administrateur :
 - ✓ gestion compte d'utilisateur.
 - ✓ gestion dossiers d'étudiant.
 - ✓ consultation de dossiers.
 - ⇒ A l'utilisateur:
 - ✓ consultation de dossiers.

2.3. Spécification des besoins non fonctionnels:

Elle s'agit des besoins qui caractérise le système. Ce sont des besoins en matière de performance. Ces besoins peuvent concerner les contraintes (langage de programmation, type SGBD, de système d'Exploitation...)

Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

- L'application devra être capable de:
 - ✓ Tourner en réseau
 - ✓ Etre compatible avec n'importe quel système d'exploitation.

2.4- Diagramme de cas d'utilisation:

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en oeuvre. Dans cette étape, il s'agira de structurer les besoins des utilisateurs et les objectifs correspondants. [1]

- Le tableau suivant récapitule l'ensemble des cas d'utilisation:

Acteur	Cas d'utilisation	
Administrateur	Gestion compte d'utilisateur	Création de compte
		Modification de compte
	Gestion de dossier d'étudiant	Inscription d'étudiant
		Mis à jour des dossiers d'étudiant
Consultation de Dossier		
Utilisateur	Consultation de dossier	

Tableau 1: Liste des cas d'utilisations.

- Dans la figure Suivante nous présentons notre diagramme de cas d'utilisation:

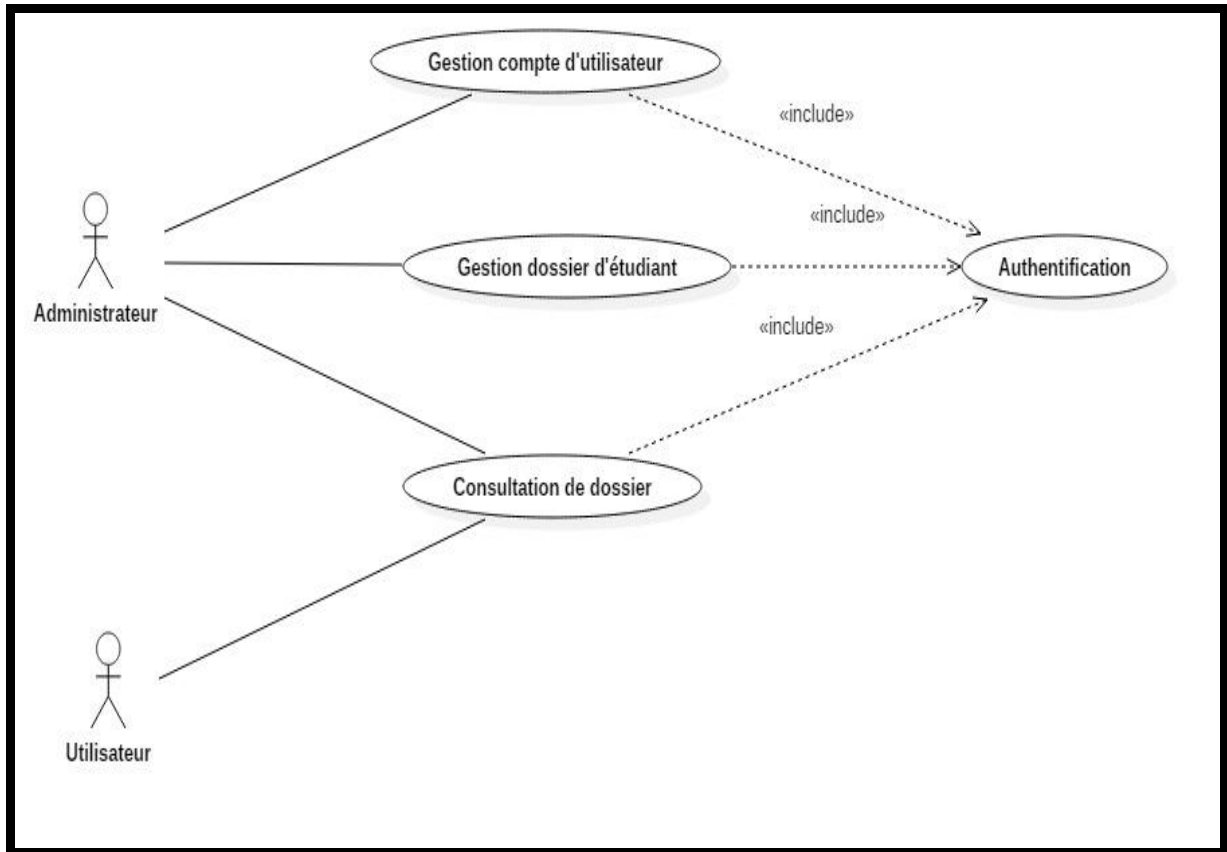


Figure 4: Le diagramme de cas d'utilisation.

3. Conclusion:

Le but de la spécification des besoins est de donner les moyens à l'utilisateur d'appréhender rapidement le fonctionnement général et de comprendre les détails de chaque fonctionnalité. Nous venons alors de présenter dans ce chapitre les cas d'utilisation de chaque acteur ainsi que les scénarios de ces utilisations de façon concise et simple afin de pouvoir entamer la conception qui fera l'objet du prochain chapitre.



Chapitre III

Conception

1. Introduction

Le but de ce chapitre est de faire une bonne réalisation à notre projet . Alors , il faut faire une conception pour garantir une longévité en terme de qualité et de rentabilité.

Il existe plusieurs méthodes et langages de conception , parmi eux figure la conception orienté objet que nous avons adopté . Le choix de cette approche se trouve comme origine dans le fait parce qu'elle caractérisée par la stabilité de la conception par rapport au monde réel.

2. Conception détaillée

2.1. Dictionnaire des données:

Désignation	Attributs	Type	Taille
Identification de profile	ID_Pr	Entier	2
Nom de profile	N_Pr	Char	20
Prenom de profile	P_Pr	Char	20
Date de naissance	D_Ns	Date	/
Lieu de naissance	L_Ns	Char	20
Sex	Sex	Char	20
Adresse	Adr	Char	20
Role	Role	Char	15
Identification d'utilisateur	ID_User	Entier	2
Nom d'utilisateur	User_N	Char	15
Mot de passe	P_W	Varchar	10
Matricule d'étudiant	Mat	Réel	10
Nom d'étudiant en "Arabe"	N_Ar	Char	20
Prenom d'étudiant en "Arabe"	P_Ar	Char	20
Nom d'étudiant en "Français"	N_Fr	Char	20
Prenom d'étudiant en "Français"	P_Fr	Char	20
Date de naissance d'étudiant	D_Ns	Date	/
Lieu de naissance d'étudiant	L_Ns	Char	20
Sex d'étudiant	Sex	Char	4
Adresse d'étudiant	Adr	Char	20
Branche de BAC	Br_BAC	Char	15
Wilaya de BAC	W_BAC	Char	15
Moyen de BAC	M_BAC	Réel	4
Année de BAC	An_BAC	Entier	4
Identification de groupe	ID_G	Entier	2

Identification de niveau	ID_Nv	Entier	2
Section de niveau	Sec	Char	15
Année de niveau	An	Date	/
Identification de spécialité	ID_Sp	Entier	2
Nom de spécialité	N_Sp	Char	20
Identification de classe	ID_Cls	Entier	2
Nom de classe	N_Cls	Char	15
Identification de département	ID_Dprt	Entier	2
Nom de département	N_Dprt	Char	25
Matricule de fichier	Mat	Réel	10
Type de fichier	T_Fch	Char	15
Nom de fichier	N_Fch	Char	20
Path de fichier	Pth_Fch	Varchar	60
Numéro de fichier	Num_Fch	Entier	4
Text de fichier	Txt_Fch	Char	150
Date de fichier	D_Fch	Date	/

Tableau 2: Dictionnaire des données.

2.2. Le diagramme de classe:

Le diagramme des classe est considéré comme le plus important de la modélisation orientée objet , il est le seul obligatoire lors d'une telle modélisation , Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe modélise les concepts du domaine d'application ainsi que les concepts internes créés de toute pièces dans le cadre de l'implémentation d'une application .Chaque langage de programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas , etc.), mais le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. [2]

- Le tableau des classes candidates de notre application :

Classes candidates	Attributs	Comportement
Profile	ID_Pr N_Pr P_Pr D_Ns L_Ns Sex Adr Role	/
User	ID_User User_N P_W	/
Administrateur	User_N P_W	Gerer_utilisateur() Gerer_Etudiant() Consulter()
Utilisateur	User_N P_W	Consulter()
Etudiant	Mat Nom Prenom D_Ns L_Ns Sex Adr Br_BAC W_BAC M_BAC An_BAC	/
Groupe	ID_G	/
Niveau	ID_Nv Sec An	/
Specialite	ID_Sp N_Sp	/
Classe	ID_Cls N_Cls	/
Departement	ID_Dprt N_Dprt	/
Fichier_dossier	Mat T_Fch N_Fch Pth_Fch Num_Fch	/

	Txt_Fch D_Fch	
--	------------------	--

Tableau 3: Table des classes candidates.

- Le diagramme de classe :

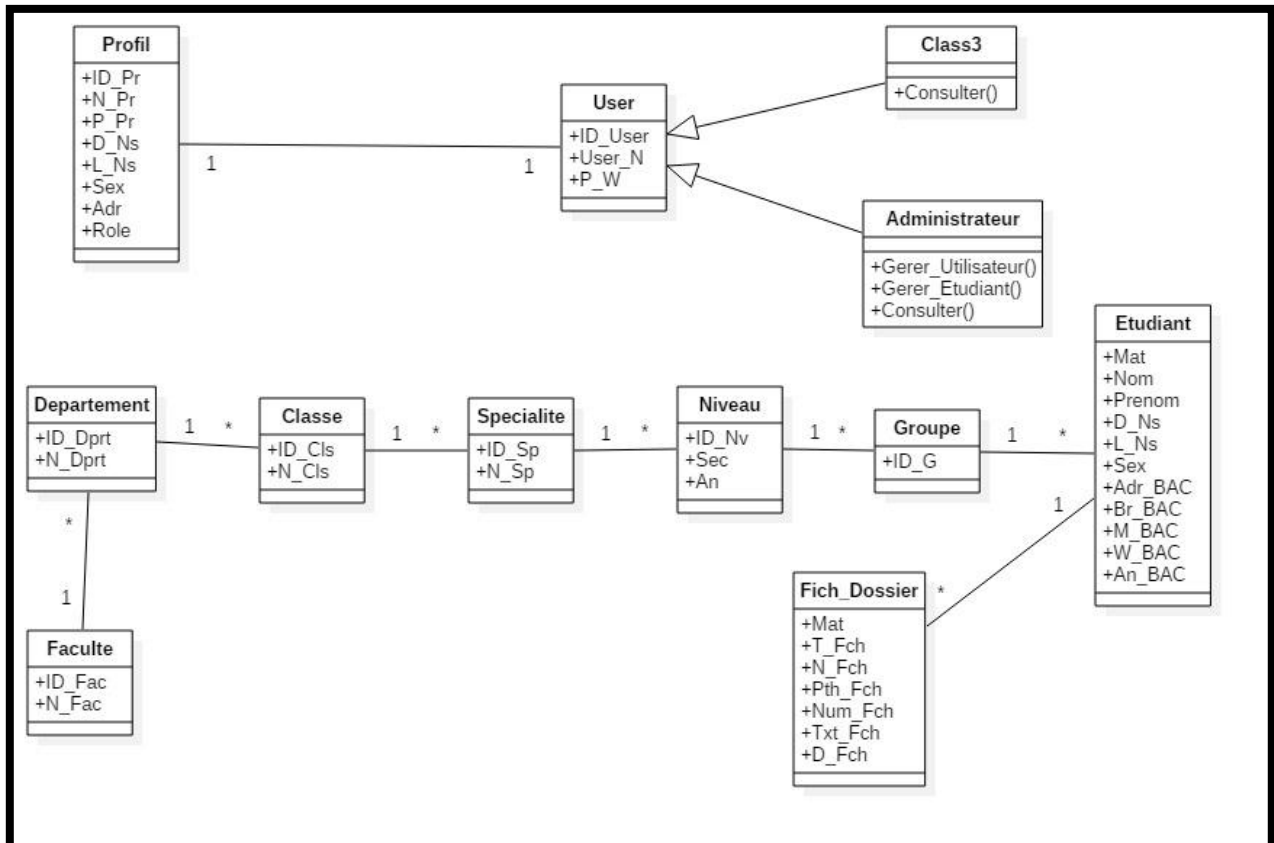


Figure 5: Le diagramme de classe.

2.3. La base des données de l'application:

- Notre base de données se compose de neuf tables:

Tables	Identifiant	Attributs
Profile	ID_Pr	ID_Pr N_Pr P_Pr D_Ns L_Ns Sex Adr Role
Utilisateur	ID_User	ID_User User_N P_W
Etudiant	Mat	Mat N_Ar P_Ar N_Fr P_Fr D_Ns L_Ns Sex Adr Br_BAC W_BAC M_BAC An_BAC
Groupe	ID_G	ID_G
Niveau	ID_Nv	ID_Nv Sec An
Specialite	ID_Sp	ID_Sp N_Sp
Classe	ID_Sp	ID_Cls N_Cls
Departement	ID_Dprt	ID_Dprt N_Dprt
Fichier_dossier	Mat	Mat T_Fch N_Fch Pth_Fch Num_Fch

		Txt_Fch D_Fch
--	--	------------------

Tableau 4: La base des données de l'application.

2.4. Les diagrammes de séquence:

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML .On montre ces interactions dans le cadre d'un scenario d'un diagramme des cas d'utilisation .Dans un souci de simplification , on représente l'acteur principal à gauche du diagramme et les acteurs secondaires éventuels à droite du système .Le but étant de décrire comment se dérouler les actions entre les acteurs ou les objets .Les périodes d'activité de classes son symbolisées par des rectangles. [3]

a- Cas d'utilisation "Authentification":

a.1- Diagramme de cas d'utilisation:

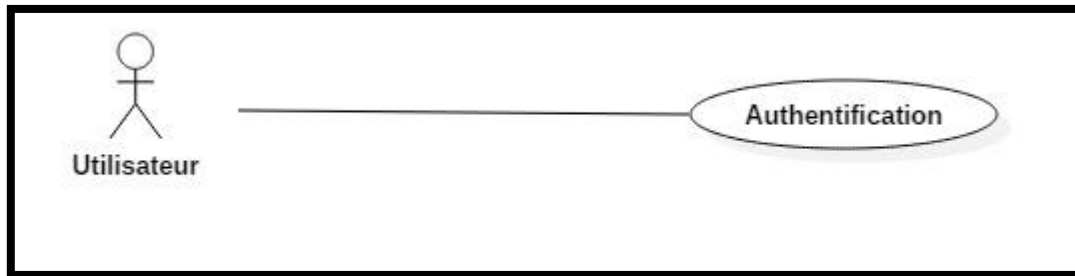


Figure 6: Diagramme de cas d'utilisation " Connexion au site ".

a.2- Diagramme de séquence:

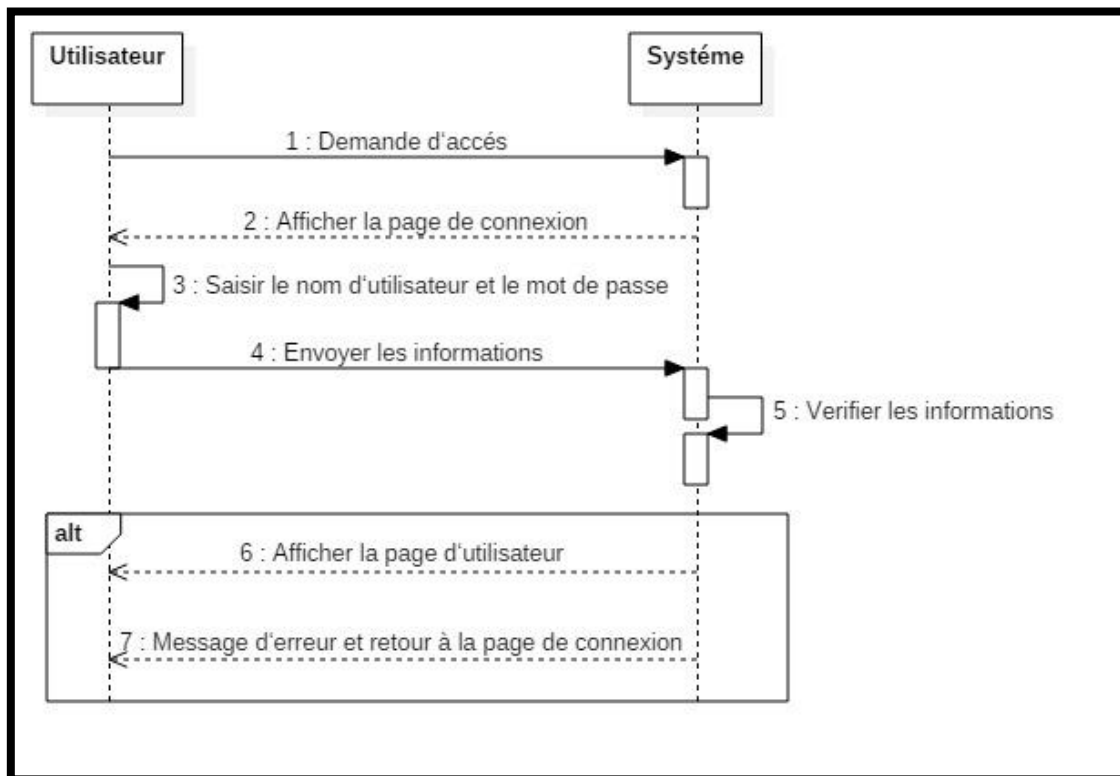


Figure 7: Diagramme de séquence " Connexion au site ".

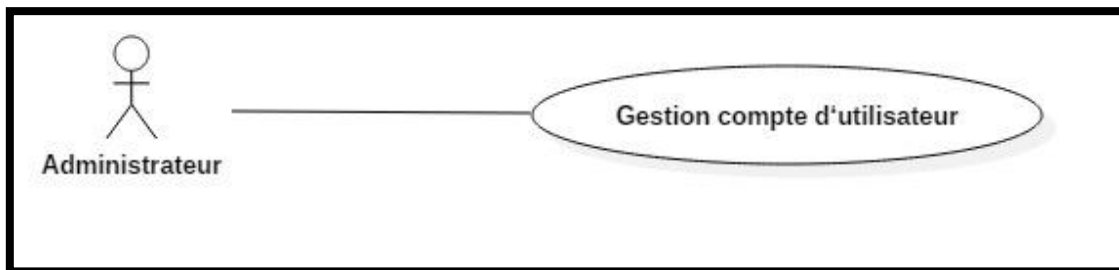
b-Cas d'utilisation "Gestion compte d'utilisateur":**b.1- Diagramme de cas d'utilisation:**

Figure 8: Diagramme de cas d'utilisation " Gestion compte d'utilisateur ".

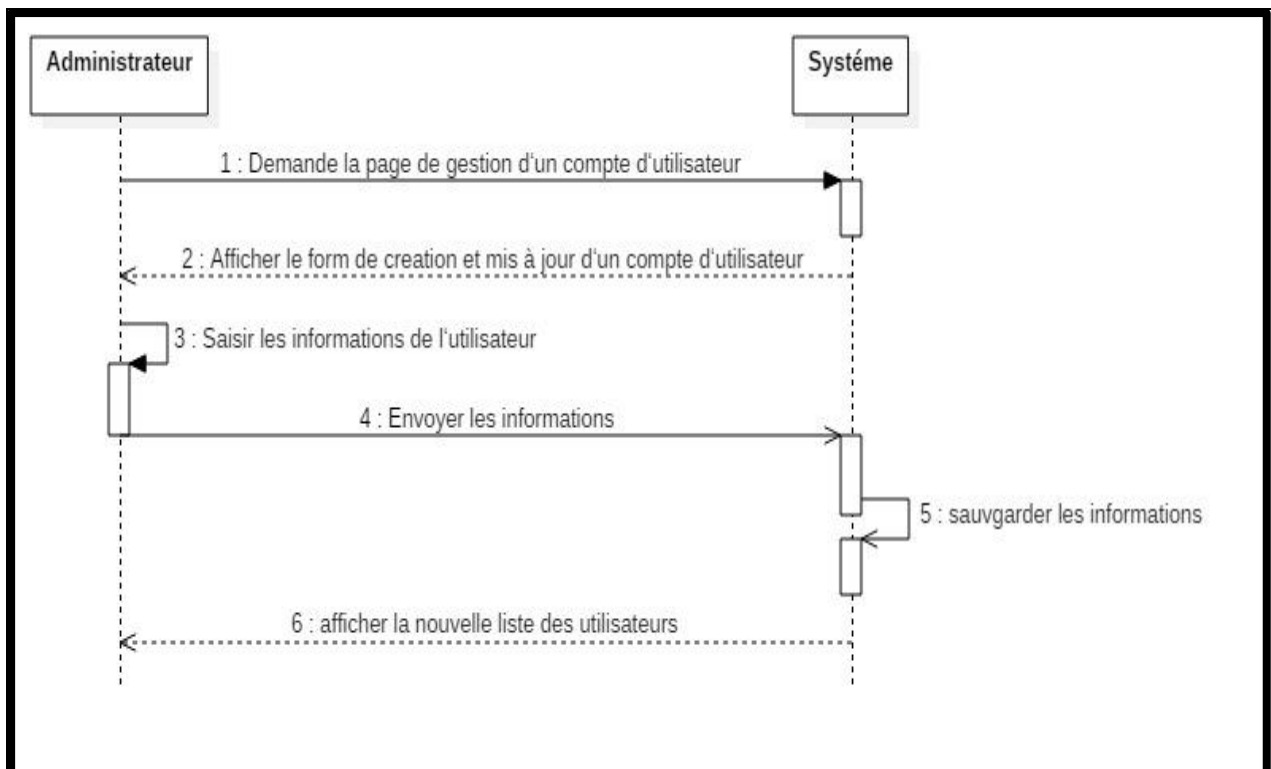
b.2- Diagramme de séquence:

Figure 9: Diagramme de séquence " Gestion compte d'utilisateur ".

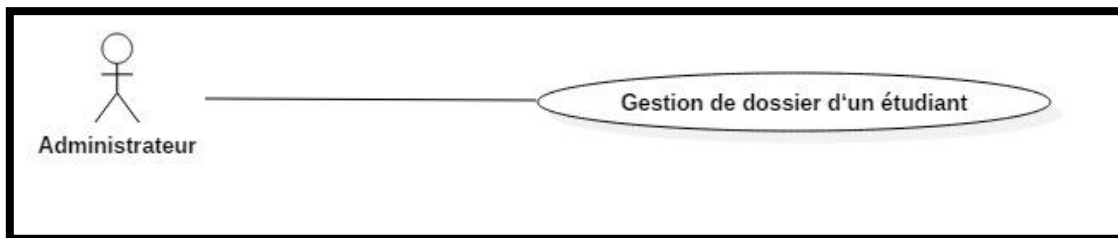
c-Cas d'utilisation "Gestion de dossier d'un étudiant":**c.1- Diagramme de cas d'utilisation:**

Figure 10 : Diagramme de cas d'utilisation " Gestion de dossier d'un étudiant ".

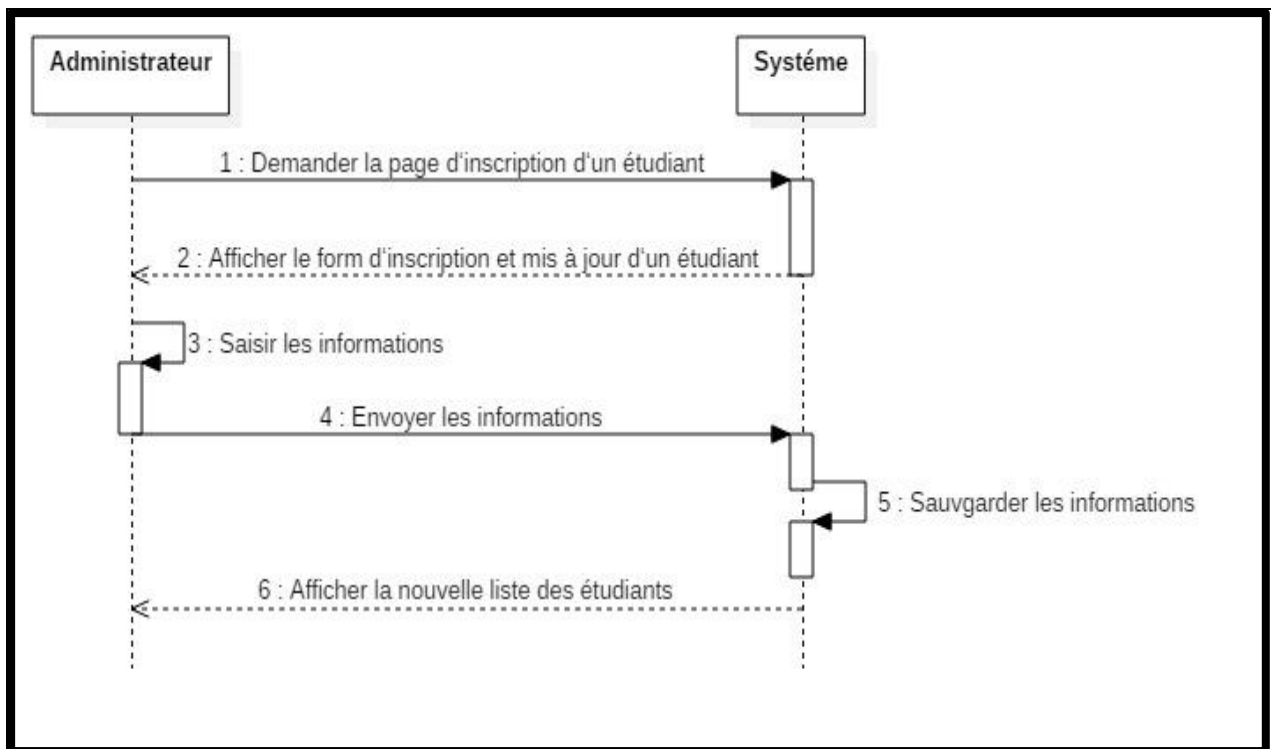
c.2.1- Diagramme de séquence:

Figure 11: Diagramme de séquence " Inscription d'un étudiant ".

c.2.2- Diagramme de séquence:

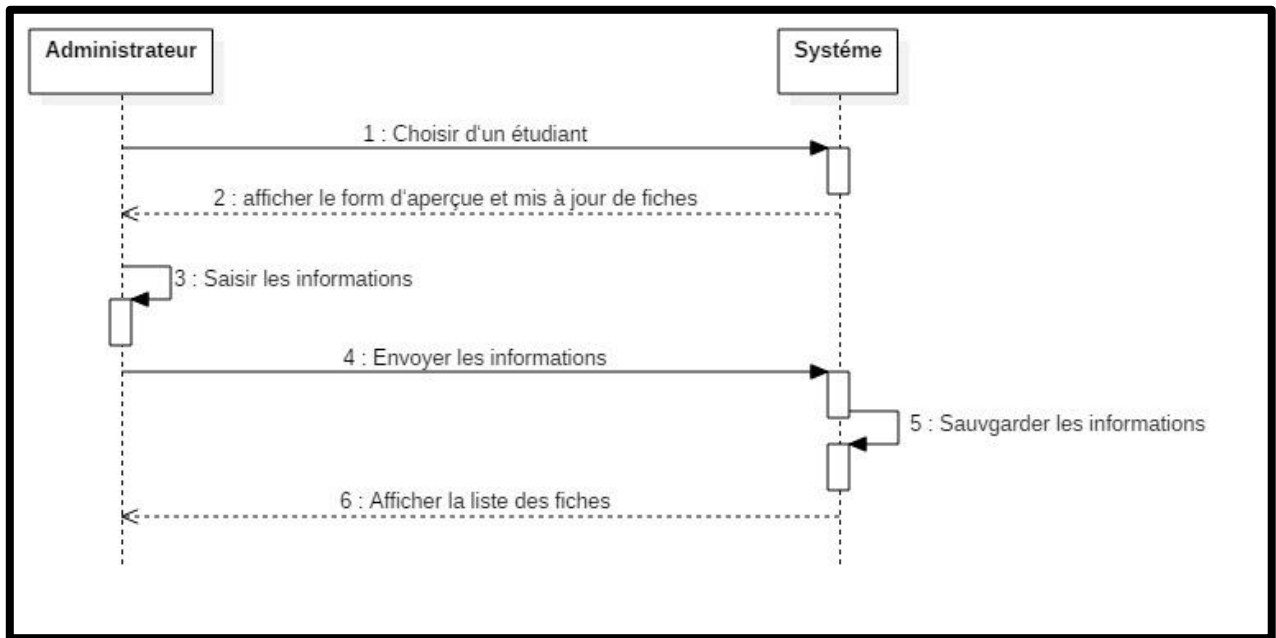


Figure 12: Diagramme de séquence " Mise à jour d'un dossier d'étudiant ".

d-Cas d'utilisation "Consultation de dossier":

d.1-Diagramme de cas d'utilisation:

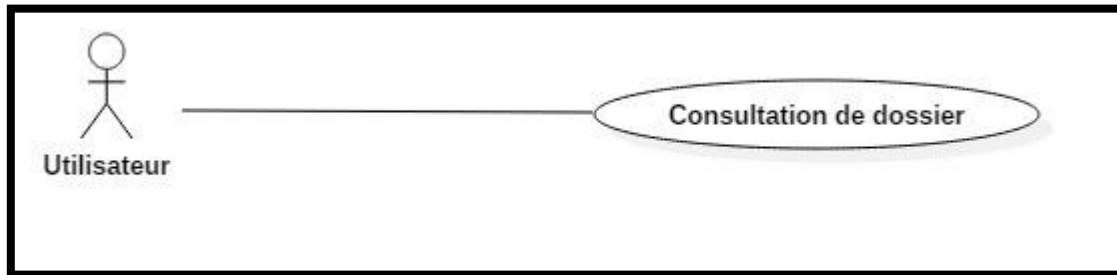


Figure 13: Diagramme de cas d'utilisation " Consultation de dossier ".

d.2-Diagramme de séquence:

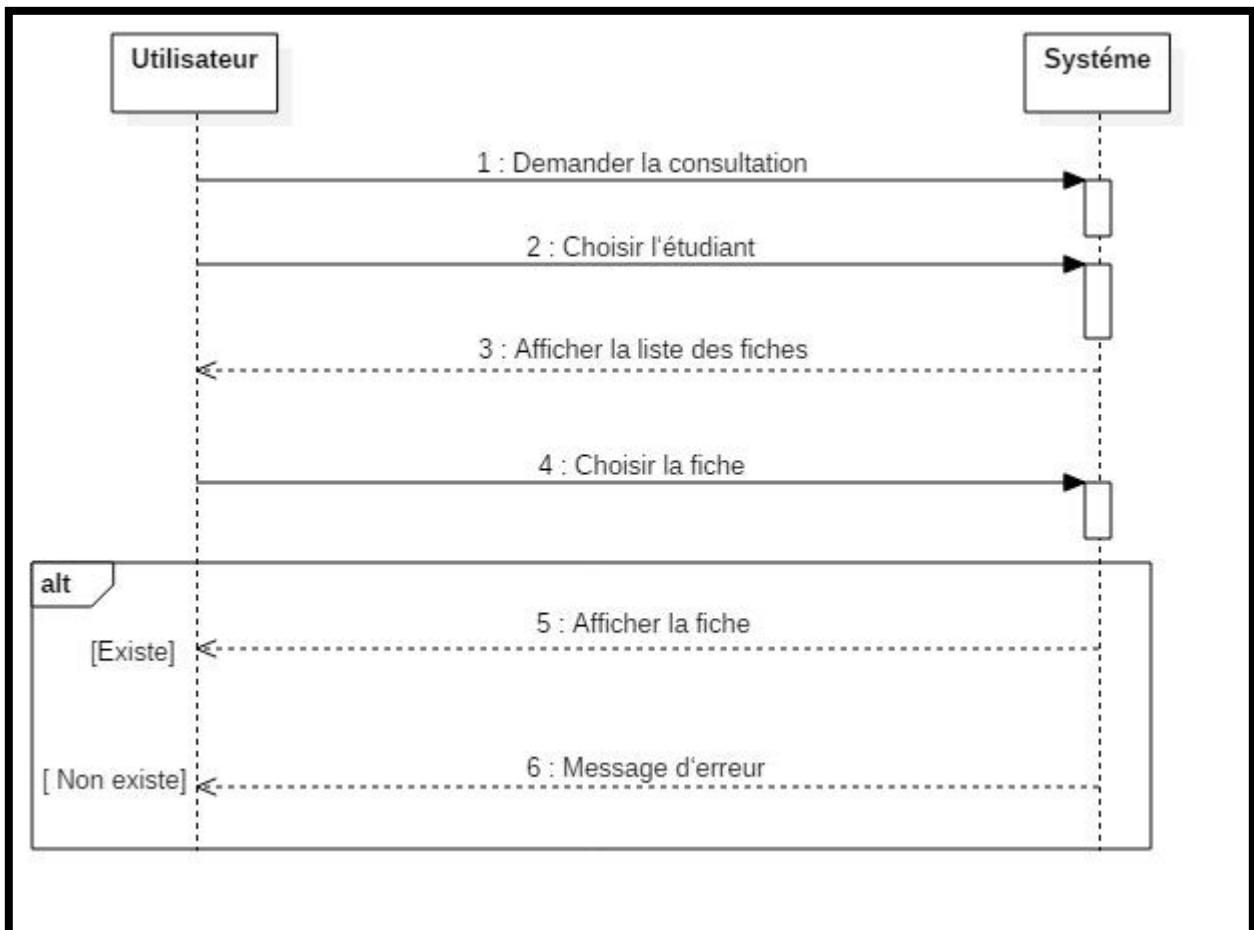


Figure 14: Diagramme de séquence " Consultation de dossier ".

3. Conclusion:

La conception, c'est une étape plus important qui précède le développement du système durant laquelle nous avons suivi une démarche de développement débutant par capturer les besoins des utilisateurs jusqu'à la codification, en passant par l'analyse et la conception du système.



Chapitre IV
Réalisation

1. Introduction

Après la présentation théorique , la détermination des besoins de l'application et la concevoir d'une manière précise , on essaye dans ce chapitre de les mettre en œuvre par une réalisation pratique d'un système , on va passer à la réalisation de site web.

Cette partie constitue la dernière étape de ce mémoire et pour objectif d'exposer le travail achevé .

Nous intéressons aux choix des outils de développement et à l'implémentation de notre application .

2. Environnement de travail

2.1 Environnement matériel:

- ✓ PC portable HP
- ✓ Processeur : Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz 1.70 GHz
- ✓ Mémoire installé : 4GB
- ✓ Système d'exploitation : 64 bits

2.2 Environnement logiciel:

Le langage C#:

Le C# est un langage de programmation orienté objet, commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET.

Il est dérivé du C++ et très proche du Java dont il reprend à la syntaxe générale ainsi que les concepts, y ajoutant des notions telles que la surcharge des opérateurs, les indexeurs et les délégués. Il est utilisé notamment pour développer des applications web sur la plateforme ASP.NET.

Caractéristiques

C# est un langage dérivé du C++, il apporte un typage sûr, ainsi que les possibilités d'encapsulation, d'héritage et de polymorphisme des langages orientés objet¹. En C# tous les types sont des objets. Le langage comporte un ramasse-miettes et un système de gestion d'exceptions.

Le typage sûr signifie notamment que les opérations suivantes sont refusées: utilisation de variable non initialisée, tentative d'accéder au-delà des limites d'un tableau, conversions de type dont les résultats ne sont pas prévisibles, dépassement des limites lors d'opérations arithmétiques.

Beaucoup de possibilités de Java se retrouvent dans C# et il y a une forte ressemblance entre un code écrit en C# et le code équivalent en Java.

En C# les variables peuvent être d'un type référence ou d'un type valeur. Les types valeur sont les types primitifs, les énumérations, les strict et les types nullable. Les types référence sont les classes, les interfaces, les tableaux et les délégante.

- **types primitifs**

Les types primitifs sont sbyte, short, int, long, byte, ushort, uint, ulong, char, float, double, decimal et bool.

- **classe**

La construction la plus fondamentale du langage C# sont les classes. Celles-ci peuvent contenir des constantes, des champs, des propriétés, des indexeurs, des méthodes, des événements, des opérateurs, des constructeurs, des destructeurs ou des sous-classes². Les classes élémentaires sont string et object.

- **struct**

les struct sont similaires aux classes, mais ce sont des types valeurs et ils ne peuvent pas être hérités.

- **delegate**

un delegate est une référence à une méthode qui comporte certains paramètres. Les delegates permettent d'assigner des méthodes à des variables et les passer en paramètre.

- **enum**

un type énuméré est un type valeur qui comporte un lot de constantes. chaque type énuméré a un type sous-jacent: un type primitif déterminé en fonction des valeurs des constantes.

- **type nullable**

Les nullable sont des types primitifs qui peuvent en plus avoir la valeur null. chaque type primitif T a un type nullable associé T?. Par exemple une variable de type int? peut contenir un int ou null. [4]

HTML:

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets. [5]

CSS:

CSS (Cascading Style Sheets:feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000. [6]

JavaScript:

JavaScript est un langage de programmation qui est exécuté par le navigateur, du côté client. Il n'entraîne donc aucune charge pour le serveur. Il permet, en se basant sur le DOM (pour « Document Object Model », ou « modèle objet de document » en français), de modifier la page HTML après qu'elle ait été envoyée par le serveur au navigateur .

JavaScript modifie le DOM, et ces modifications sont ensuite répercutées sur la page affichée à l'utilisateur .

Le DOM est une représentation objet de la page web. Ainsi, à chaque balise HTML correspond un objet en mémoire, et le document est représentable sous forme d'arbre. Un élément peut donc avoir un parent, des ascendants, des descendants et des frères. [7]

2.2.1 Langage de programmation (IDE)

2.2.1.1 Présentation

Microsoft Visual Studio:

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle Visual Studio 2015.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

Langages supportés par Visual Studio

- ✓ Visual Basic
- ✓ Visual C++
- ✓ Visual C#
- ✓ Visual Web Developer
- ✓ SQL Server
- ✓ C [8]

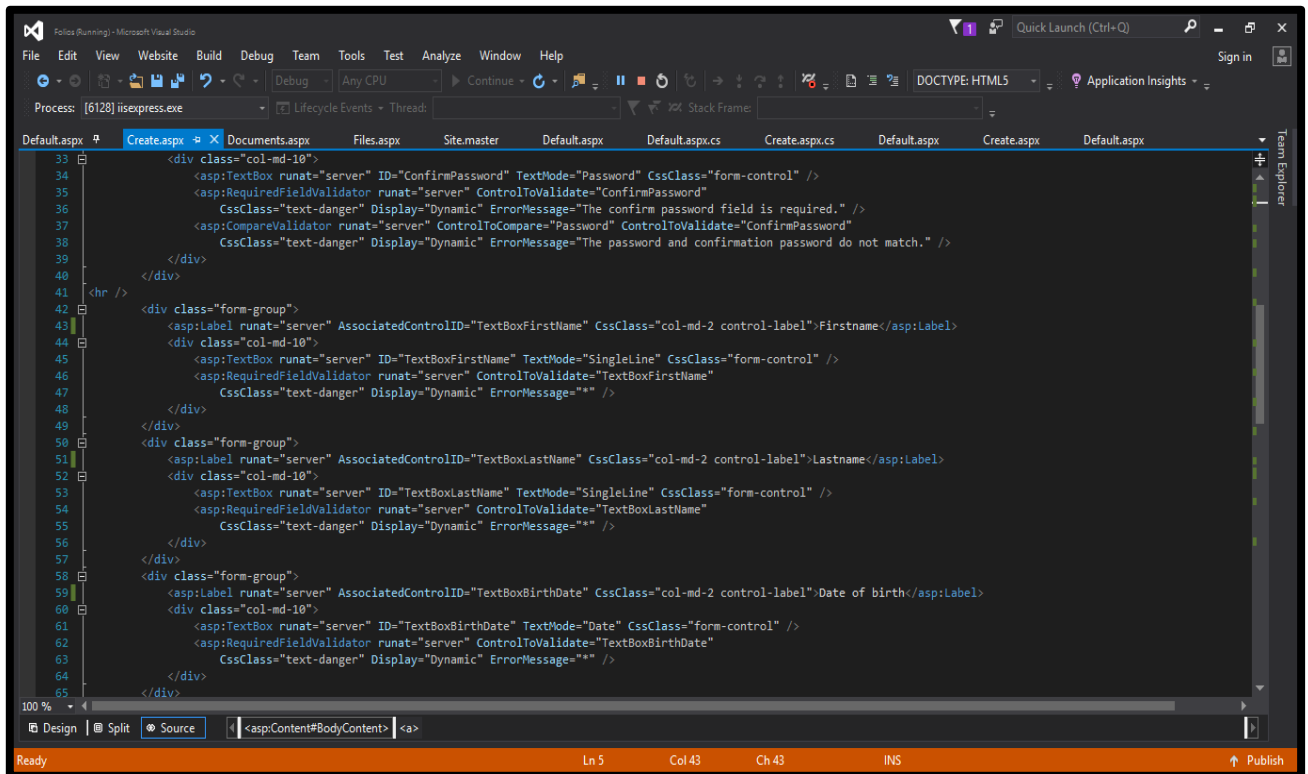


Figure 15: Interface de microsoft visual studio.

2.2.1.2 Système de gestion de base de données utilisé:

Microsoft SQL Server 2012 est un système de gestion de base de données relationnelles (SGBDR) conçu pour l'environnement de l'entreprise. Comme ses prédécesseurs, SQL Server 2012 comprend un ensemble d'extensions de programmation pour améliorer la Structured Query Language (SQL), un langage interactif et programmation standard pour obtenir des informations à partir et mettre à jour une base de données.

Microsoft SQL Server 2012, qui supprime SQL Server 2008 R2, offre de nouvelles fonctionnalités, notables parmi eux ce qui suit.

- ✓ index columnstore: lecture seule index que les données de groupe, de rationaliser le traitement des grandes questions d'entrepôt de données.
- ✓ Prise en charge de Windows Server Core: Ceci est une version allégée qui place une demande beaucoup plus faible sur les ressources informatiques d'une installation complète.

- ✓ PowerView: Permet de générer des mash-ups de business intelligence (BI) rapports.
- ✓ Audit améliorée: Les utilisateurs peuvent personnaliser leurs journaux d'audit pour accueillir un plus large éventail d'événements avec une plus grande flexibilité que ce qui était possible auparavant.
- ✓ AlwaysOn: Les utilisateurs peuvent échouer sur plusieurs bases de données et de lire des copies secondaires, l'amélioration de reprise après sinistre (DR) opérations.
- ✓ Distributed Replay: Une charge de travail peut être prise à partir d'un serveur de production et joué sur un autre serveur pour le tester dans des conditions réalistes avant de le déployer.
- ✓ Le code SQL Server d'origine a été développé par Sybase. À la fin des années 1980, Microsoft, Sybase et Ashton-Tate ont collaboré pour produire la première version du produit, SQL Server 4.2 pour OS / 2. Par la suite, Sybase et Microsoft ont offert des produits SQL Server.
- ✓ Cette dernière mise à jour en Novembre 2012 [9]

2.2.1.3 Server d'application

ASP.NET est une technique pour générer à la demande des pages web, lancée par Microsoft en juillet 2000, et utilisée pour mettre en œuvre des applications web. Il s'agit d'une évolution majeure d'Active Server Pages (abr. ASP), par laquelle cette technique a été incorporé dans la plateforme Microsoft .NET.

Le moteur d'ASP.NET est un filtre branché sur le service web Internet Information Services (abr. IIS). Il est distribué avec le framework .NET. ASP.NET peut être utilisé avec n'importe quel langage de programmation pour la plateforme .NET - Visual Basic.NET, C#, JScript...

Principes

ASP.NET est une technique pour générer des pages web dynamiques. Une page ASP.NET est composée de deux parties: d'un côté du HTML, et de l'autre des instructions de programme. Ces instructions sont utilisées pour générer le résultat d'une demande de page qui sera envoyée au navigateur web. Le résultat que reçoit le navigateur est du HTML ordinaire. Une page ASP.NET comporte des Web controls: des portions de HTML qui peuvent être modifiées par programmation.

Le moteur d'ASP.NET est un filtre, branché sur le service web Internet Information Services (abr. IIS) par son interface de programmation - ISAPI. Le moteur ASP.NET est distribué avec le framework .NET. Il peut être utilisé avec n'importe quel langage de programmation pour la plateforme .NET - Visual Basic .NET, C#, JScript... Les sites web ASP.NET sont couramment développés en utilisant un serveur web simplifié, installé sur l'ordinateur personnel du développeur.

Programmation

La programmation sur ASP.NET est orientée événement, un événement = quelque chose s'est passé. Dans ce style de programmation le système attend qu'il se passe quelque chose, par exemple que l'utilisateur presse sur un bouton. Une fonction est associée à cet événement, et cette fonction sera exécutée automatiquement par le système lorsque l'utilisateur effectue l'action. En ASP.NET la page web et la plupart des widgets peuvent provoquer des événements. Des événements correspondent également au démarrage de l'application ou à la connexion d'un utilisateur. Des événements peuvent correspondre à des actions effectuées sur le navigateur web, qui peuvent être parfois traités avec du retard, du fait qu'ASP.NET est exécuté par le serveur web.

ASP.NET comporte plusieurs lots de widgets (controls) qui peuvent être utilisés dans les pages web:

- ✓ Html controls: les balises HTML correspondant à des widgets tels que boutons d'action, des boutons radio ou des cases à cocher.
- ✓ Html server controls: il s'agit des balises HTML qui peuvent être manipulées par programmation. De nombreuses balises html peuvent être converties en Html server control en leur ajoutant l'attribut `runat="server"`.
- ✓ ASP Web controls: ils offrent les mêmes possibilités que les balises HTML, mais une programmation plus simple. Les balises pour ces widgets commencent par les lettres `asp:`. exemple: `<asp:Button ... runat="server" />`. Ces widgets sont transformés en HTML par ASP.NET avant d'être envoyés au navigateur web. Le résultat de la transformation peut différer selon le navigateur utilisé.

- ✓ Parmi les ASP Web controls, il y en a qui permettent d'afficher des tableaux contenant des données obtenues par programmation et d'autres qui permettent d'afficher un calendrier ou l'utilisateur peut choisir une date.
- ✓ Validation controls: ils offrent la possibilité d'effectuer des vérifications des actions de l'utilisateur.
- ✓ User controls et server controls: Il s'agit de widgets créés par les développeurs pour leurs besoins.

Fonctionnement

- ✓ Le pipeline d'ASP.NET
- ✓ Le moteur d'ASP.NET est un filtre, branché sur le service web Internet Information Services (abr. IIS) par son interface de programmation -ISAPI. Le filtre - une DLL - est chargé en mémoire à la première utilisation, puis pour chaque demande de page web il lit un modèle de page, puis exécute des instructions qui vont altérer le contenu de la page et renvoie le résultat à IIS, qui le transmet au client.
- ✓ L'interface ISAPI est également utilisée pour d'autres langages web tels que PHP ou Perl. Cette interface utilise la technologie COM. Microsoft fournit très peu de documentation à son sujet.
- ✓ Le moteur ASP.NET fonctionne avec un pipeline de traitement des requêtes web: Lorsque l'utilisateur navigue sur un hyperlien, une demande est envoyée au serveur web IIS. La demande traverse ensuite le pipeline d'ASP.NET jusqu'au point de sortie ou un développeur d'application peut attacher le code nécessaire au traitement de cette demande. Le moteur est indépendant du serveur web IIS.

Lorsque le serveur web reçoit une demande de page web, il détermine tout d'abord s'il s'agit de contenu statique - enregistré dans un fichier - ou de contenu dynamique - généré sur demande⁴. L'examen tient compte de l'extension du fichier. Selon la configuration, les fichiers qui se terminent par .aspx, .asax, .ascx, .ashx, .asmx, .axd ou .svc seront transmis à ASP.NET via aspnet_isapi.dll, une bibliothèque branchée à l'interface ISAPI de IIS. La

requête sera envoyée à un récepteur différent selon son extension. ASP.NET exécute ensuite un programme, puis le résultat du programme est renvoyé au serveur web qui le transmettra au navigateur web à l'origine de la demande.

- ✓ Les pages web (.aspx) ainsi que des service web (.asmx) sont mis en œuvre en utilisant deux récepteurs intégrés à ASP.NET. Beaucoup de développeurs d'application se servent uniquement de ces deux récepteurs et le fait qu'il est possible pour les développeurs d'application de brancher leur propre récepteur est souvent inconnu ou pas maîtrisé. Le récepteur pour les fichiers. asmx est cependant dépassé par Windows Communication Foundation, disponible depuis la version 3.0 du framework .Net.**[10]**

3. Interfaces de l'application



Figure 16: Interface de " Page d'accueil " .

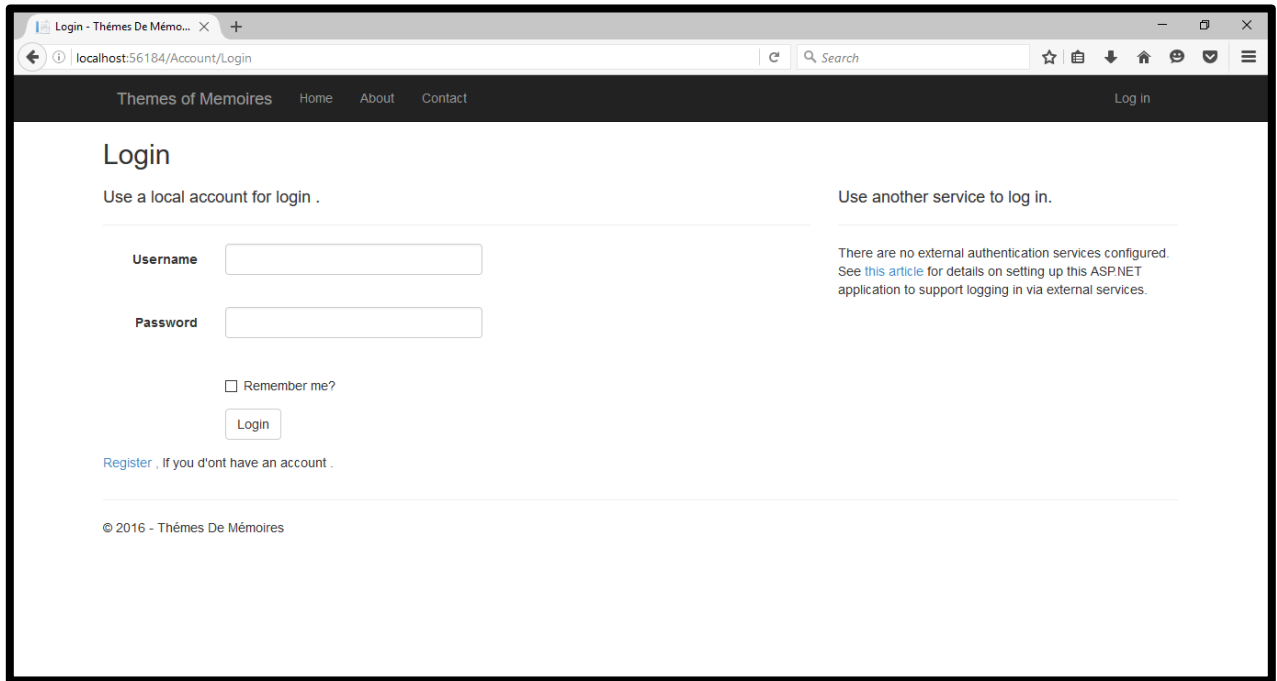
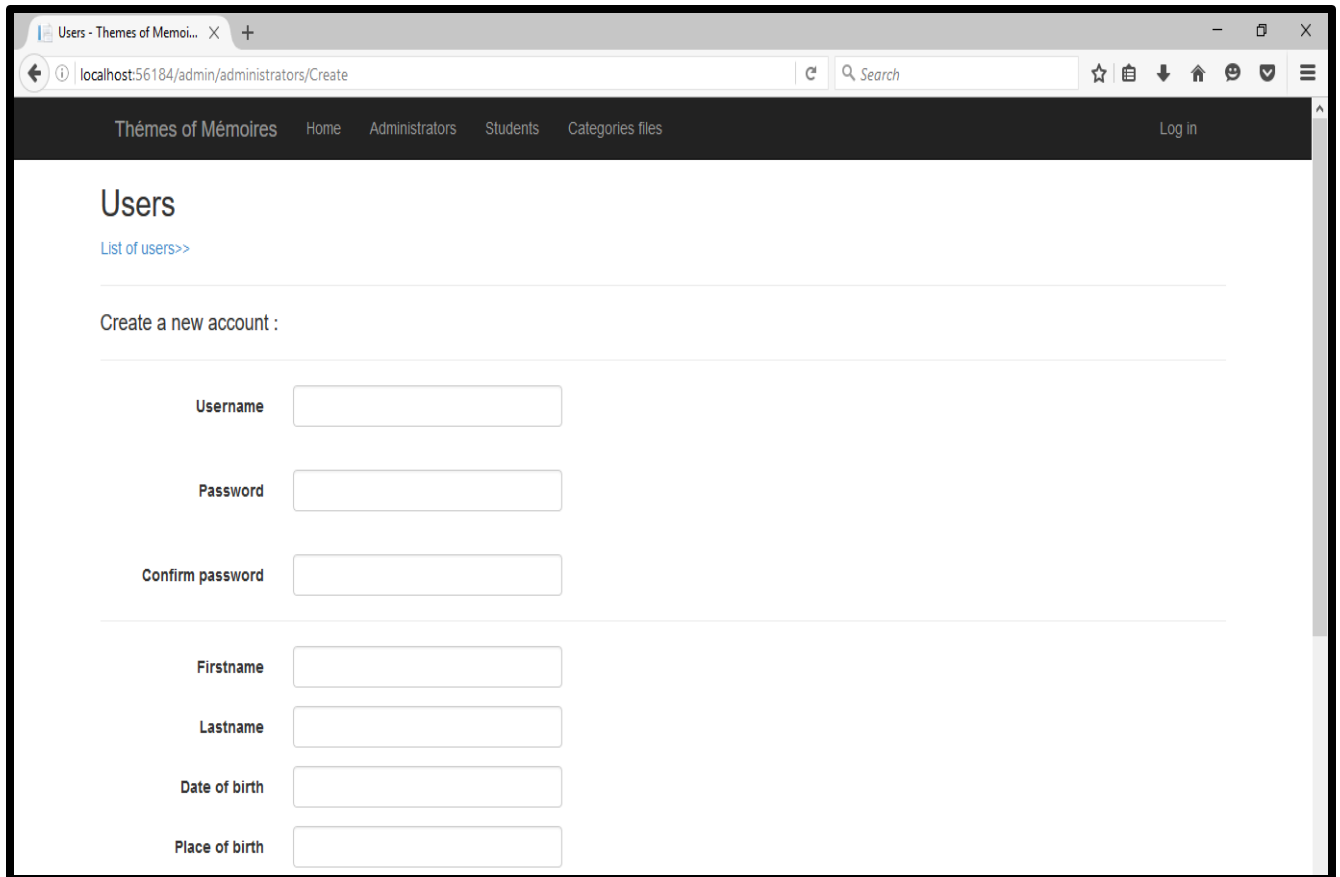


Figure 17: Interface de " Log in " .

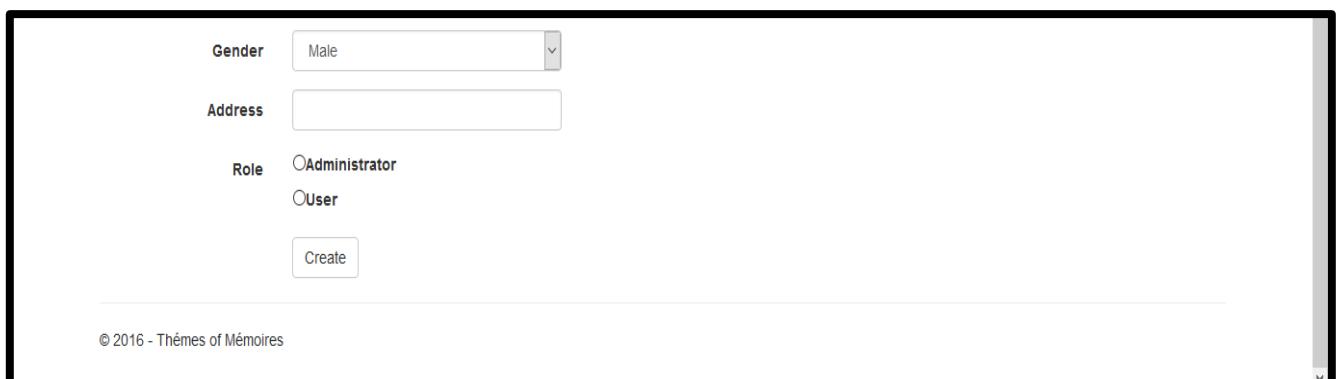
Cette interface représente l'interface de création d'un compte pour l'utilisateur, il contient le nom, le mot de passe, date de naissance, Place de naissance.



The screenshot shows a web browser window with the following elements:

- Browser tab: Users - Themes of Memoi... X +
- Address bar: localhost:56184/admin/administrators/Create
- Navigation icons: back, forward, search, star, home, refresh, checkmark, menu
- Page header: Thèmes of Mémoires | Home | Administrators | Students | Categories files | Log in
- Section title: Users
- Link: [List of users>>](#)
- Form title: Create a new account :
- Form fields:
 - Username:
 - Password:
 - Confirm password:
 - Firstname:
 - Lastname:
 - Date of birth:
 - Place of birth:

Cette interface représente l'interface de création de compte, il contient sex, adresse.



The screenshot shows a web browser window with the following elements:

- Form fields:
 - Gender:
 - Address:
 - Role: Administrator, User
- Button:
- Page footer: © 2016 - Thèmes of Mémoires

Figure 18: Interface de " Création d'un compte".

Students - Thèmes of Me... x +

localhost:56184/admin/Students/Create

Thèmes de Mémoires Home Administrators Students Categories files Log in

Students

[List of students>>](#)

Register student

Matricule

Firstname Ar

Lastname Ar

Firstname Fr

Lastname Fr

Date of birth

Place of birth

Gender

Address

Division of BAC

State of BAC

Rate of BAC

Year of BAC

Faculty

Class

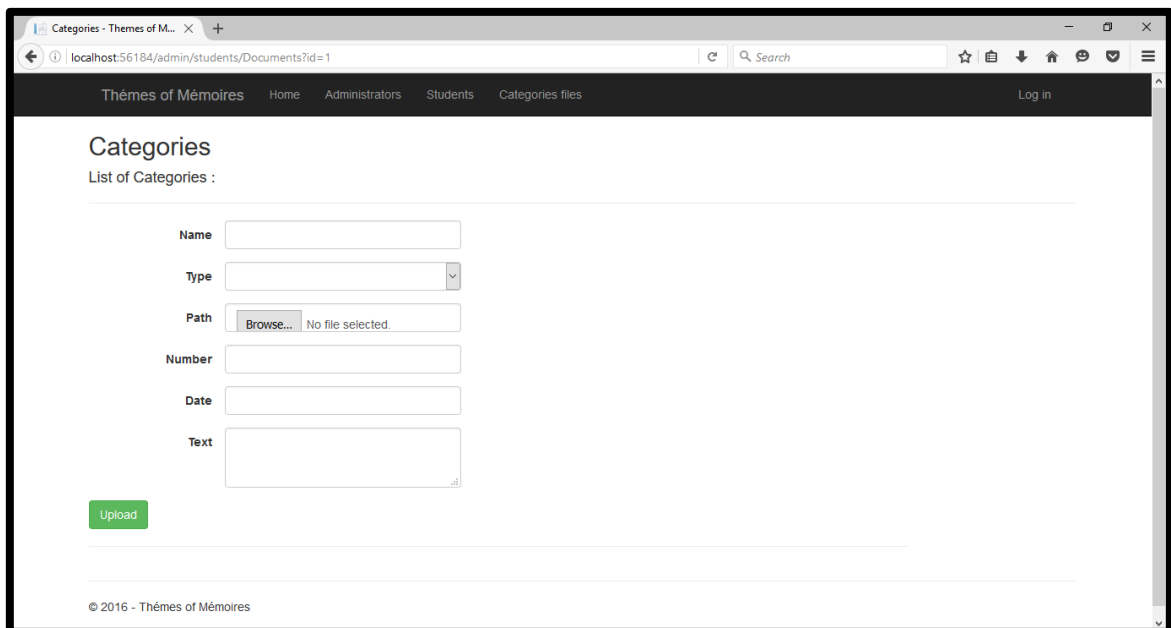
Level

Group

Univ-Year

© 2016 - Thèmes de Mémoires

Figure 19: Interface de " Inscription d'un étudiant".



The screenshot shows a web browser window with the address bar displaying "localhost:56184/admin/students/Documents?id=1". The page title is "Categories - Themes of M...". The navigation menu includes "Thèmes de Mémoires", "Home", "Administrators", "Students", "Categories files", and "Log in". The main content area is titled "Categories" and contains the text "List of Categories :". Below this, there is a form with the following fields:

- Name:
- Type:
- Path: (with a "Browse..." button)
- Number:
- Date:
- Text:

A green "Upload" button is located below the form fields. At the bottom of the page, there is a copyright notice: "© 2016 - Thèmes de Mémoires".

Figure 20: Interface de "Update fiche".

4. Conclusion :

Dans ce chapitre , nous avons décrit le déroulement dans notre projet , ensuite nous avons spécifié les langages de programmation choisis. Puis nous avons exposé les interfaces principales de notre application. Ces interfaces présente tous les fonctionnalités nécessaires pour l'affectation des projets de fin d'étude .

Conclusion générale

L'objectif de notre travail était le Développement d'une Web de gestion des dossiers des étudiants dans toutes les institutions .

Nous avons commencé notre travail par une introduction générale pour avoir une vue globale de situation du système existant , Par la suite , Dans le premier chapitre , nous avons passé par l'étude de l'existant et on la chapitre et proposé des solutions .

Dans le deuxième chapitre , on a passé par l'analyse et la spécification des besoins et on a désigne notre diagramme de cas d'utilisation .

Dans le troisième chapitre , on a présente la conception détaillée de notre projet et on a désigne notre diagramme de classe .

Dans le dernier chapitre , on a passé par la réalisation de notre projet en utilisant des différents langages de programmation (C# ,HTML ,CSS, , Java Script) et un système de gestion de base des donnée (Microsoft SQL Server 2012) avec un serveur (ASP.NET).

Ce projet nous a été très bénéfique, car il nous permet de concrétiser nos connaissances théoriques issues des trois années d'étude , de connaitre de près l'environnement de programmation et d'acquérir de nouvelles connaissances.

نقترح في مشروع تخرجنا إنجاز برنامج يساعد الجامعة في تطوير عملية تنظيم نماذج ملفات الطلبة وحفظها بطريقة آلية, باعتبارها واحدة من أهم الأعمال التي لها تأثير مباشر على الجامعة والتي لا تقل أهمية عن باقي الأعمال الأخرى, مما يوفر ربح الوقت والسرعة من ناحية البحث واستخراج نسخ عنها تحوي البيانات اللازمة بعد دمجها من مصدر بيانات كقاعدة بيانات عند الحاجة إلى ذلك. لهذا السبب اخترنا أن نركز عملنا على طريقة التطوير والنمذجة من أجل ضمان الاستكمال الناجح لمشروعنا.

الكلمات المفتاحية: UML. SQL.SGBD

Résumé

Nous proposons dans notre projet d'élaborer un programme qui peut aider l'université pour développer les procédures d'organiser les modèles des dossiers des étudiants et de les sauvegarder d'une façon automatique et rapide parce que ces procédure sont parmi les procédures les plus importantes qui ont une influence directe sur l'établissement ce que permet de gagner le temps et la rapidité de rechercher et de extrait les documents avec les données que nous avons besoin.

C'est pour ça que nous avons choisi d'axer notre travail sur l'application d'une méthode de développement et de modélisation pour assurer et garantir une bonne réalisation de notre projet.

Mots clés: UML. SQL.SGBD

Abstract

We suggest on our project completion a program that helps university to develop process organization of students documents and save them in an automated fashion and fast, as one of the most important works that have a direct impact on the institution which is not less important than the rest of the other works, which saves time and won the speed of hand search and extraction when needed.

For this reason, we choose to focus our project on the way of development and modeling in order to ensure the successful completion of our project.

Key words : UML. SQL.SGBD