

## *Projet de Fin d'Etudes*

# Application web de gestion des classes

Réalisé par

**Aya Moufid**

**Chaimaa Souhir**

**Kawtar Farrouji**

Encadré par

**Mr Chbihi Reda**

Le " /06/2022"

### Membres du Jury

Fsac

Président

Fsac

Rapporteur

Fsac

Examinateur

Chbihi Reda

Fsac

Encadrant

Département de Mathématiques et informatique

Licence Sciences Mathématiques et informatique

# Dédicace



*Nous tenons à dédier, du profond de nos coeurs ,  
ce travail de fin d'étude à :*

- Nos chers parents ,nos frères et sœurs pour leurs soutiens inconditionnels et leurs encouragements incessants.*
- Notre encadrant Mr REDA CHBIHI qui nous a permis de mener à bien ce travail.*
- Nos enseignants pour les efforts qu'ils ont déployés pour nous permettre d'acquérir un niveau élevé de formation .*
- Tous ceux que nous connaissons de près ou de loin.*

# **Remerciement**



**C'est avec un réel plaisir qu'on exprime ici nos reconnaissances à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail. Nous remercions tout d'abord Allah le tout puissant de nous avoir permis de mener à terme ce projet qui présente pour nous le point de départ de notre carrière.**

**Ainsi on adresse les sincères remerciements à notre cher encadrant de projet de fin d'étude Mr. REDA CHBIHI, pour leurs conseils précieux, leur soutien ,leur disponibilité et pour ses directives, qui nous a aidé tout au long de l'élaboration de ce projet et qui a tout fait pour nous mettre dans les bonnes conditions de travail. Nos remerciements vont aussi à l'encontre de nos chers parents qui n'ont ménagés aucun effort et qui ne cesseront de le faire, pour nous soutenir à tout moment de nos vie.**

**N'oubliant pas nos remerciements à tout le corps professoral de la Faculte des sciences Ain Chok pour la qualité de son enseignement ainsi sa disponibilité tout au long de ces trois années de formation.**

# Liste des acronymes



<b>UML</b>	<b>UNIFIED MODELING LANGUAGE</b>
<b>MCD</b>	<b>CONCEPTUAL DATA MODEL</b>
<b>MPD</b>	<b>MODÈLE PHYSIQUE DE DONNÉES</b>
<b>MVC</b>	<b>MODEL VIEW CONTROL</b>
<b>CSS</b>	<b>CASCADING STYLE SHEETS</b>
<b>HTML</b>	<b>HYPertext MARKUP LANGUAGE</b>
<b>JS</b>	<b>JAVASCRIPT</b>
<b>PHP</b>	<b>HYPertext PREPROCESSOR</b>
<b>BD</b>	<b>BASE DE DONNÉES</b>
<b>SQL</b>	<b>STRUCTURED QUERY LANGUAGE</b>

# Résumé



**Dans le cadre de notre formation en licence Mathématique Informatique et afin d'appliquer nos connaissances théoriques requis lors de cette formation, nous sommes amenés à réaliser un Projet de Fin d'Etude.**

**Notre mémoire se concentre en premier lieu, sur l'étude, la conception et la réalisation d'une application web qui permet aux différents enseignants une gestion simple et efficace des classes et même pour les étudiants une utilisation simple et un accès facile pour pouvoir accéder aux différentes fonctionnalités d'une classe.**

**Toutes les étapes ont été respectées allant de l'étude préalable du plateforme jusqu'à l'implémentation en passant par l'analyse et la conception des données. Ainsi nous avons opté d'abord pour la modélisation sous UML, puis la conception des données à travers MCD, et enfin la méthode MVC qui permet l'implémentation du code.**

**Ce présent document rapporte nos missions accomplies durant ce Projet de Fin d'Etude.**

# **Abstract**



**As part of our training in Computer Science and Mathematics Bachelor and to apply our theoretical knowledge requested during this training, we are led to accomplish a Project of End of Study.**

**Our memory concentrates above all, on study comprehension and realization of an application web which allows to the different teachers a simple and efficient management of classes and even for the students an easy use to be able to achieve the different functionalities of a class.**

**All stages were respected going of prior study of platform up to the realization by way of the analysis and the comprehension of data. So we opted for modelling under UML, then comprehension of data across MCD, and finally first method MVC which allows the realization of code.**

**This present document brings back our missions fulfilled during this Plan of the End of Study.**

# Sommaire



Dédicaces.....	2
Remerciement.....	3
Liste des acronymes.....	4
Résumé.....	5
Abstract.....	6
Listes des figures.....	9
Listes des tableaux.....	11
Introduction générale.....	12

## **Chapitre I :Contexte Générale**

Introduction.....	13
Objectifs.....	14
Missions du projet.....	14
Problématique.....	15
Livrables.....	16
Risques.....	16
Méthodologies de travail.....	17
Cycle de vie.....	18
Planning prévisionnel du projet.....	20
Planning réel du projet.....	21
Conclusion.....	22

## **Chapitre I I:Analyse et conception**

Introduction.....	23
Besoins fonctionnels.....	24
Besoins non fonctionnels.....	26
Conception.....	27

### **Diagramme de Cas d'utilisation**

1-Definition.....	28
2-Description textuelle.....	29

### **Diagramme de sequence**

1-Definition.....	35
2-Diagramme de sequence : Authentification.....	36
3-Diagramme de sequence : Traitement de demande d'accès.....	37
4-Diagramme de sequence : Gestion des devoirs.....	38

### **Diagramme de classe**

1-Definition.....	40
2-Diagramme de classe.....	40

### **MCD**

1-Définition.....	43
2-Modèle conceptuel de données.....	43

Structure de la base de données.....	45
MOCKUPS.....	46
conclusion.....	56

# Bibliographie



## Chapitre III :Application

Introduction.....	57
-------------------	----

### **I. Environnement de travail**

1. Environnement Hard.....	58
2.LANGUAGES ET METHODES DE PROGRAMATION.....	58
3.Design & Multimédia.....	59
4. Outils utilisés.....	59
Introduction au modèle MVC.....	62
Modèle.....	62
Vue.....	63
Contrôleur.....	63

### **II. Principales interfaces graphiques**

Principales interfaces graphiques.....	63
Conclusion .....	72
Conclusion Générale.....	73
Bibliographie.....	74

# Liste des figures

**Figure 1 : Cycle en V**

**Figure 2 : Cycle de vie d'une application**

**Figure 3 : Diagramme de GANTT prévisionnel**

**Figure 4 : Diagramme de GANTT réel**

**Figure 5 : Diagramme de cas d'utilisation**

**Figure 6: Diagramme de séquence Authentification**

**Figure 7: Diagramme de séquence Traitement de demande d'accès**

**Figure 8: Diagramme de séquence Gestion des devoirs**

**Figure 9 : Diagramme de classe**

**Figure 10 : Modèle conceptuel de données**

**Figure 11 : Modèle physique de données**

**Figure 12 :Authentification**

**Figure 13 :mot de passe oublié**

**Figure 14 : Inscription d'un nouveau enseignant**

**Figure 15 :Inscription d'un nouveau étudiant**

**Figure 16 : la page principale de l'enseignant**

**Figure 17 :la page principale de l'étudiant**

**Figure 18 :la pages des classes de chaque étudiant**

**Figure 19 : Partie annonces**

**Figure 20 : partie documents**

**Figure 21: Affichage des documents par type**

**Figure 22: partie devoirs**

**Figure 23: partie ajout des devoirs étudiant**

**Figure 24: formulaire d'ajout d'une classe**

**Figure 25: partie annonces enseignant**

**Figure 26: partie document enseignant**

# Liste des figures



**Figure 27: partie devoirs enseignant**

**Figure 28: partie absences**

**Figure 29: partie notes**

**Figure 30: partie demandes**

**Figure 31: MVC**

**Figure32: interface d'accueil**

**Figure33 : interface d'inscription**

**Figure 34:Authentification**

**Figure35: interface page principale de l'enseignant**

**Figure36: interface gérer les annonces**

**Figure37: interface gérer les documents**

**Figure38: interface gérer les devoirs**

**Figure39: interface gérer les demandes**

**Figure40: interface gérer les absences**

**Figure41: interface gérer les notes**

**Figure42: interface d'inscription étudiant**

**Figure43: interface page principale de l'étudiant**

**Figure44: interface annonces**

**Figure45: interface document**

**Figure46: interface devoirs**

# Liste des tableaux



Tableau 1-1:Description textuelle du cas d'utilisation «S'authentifier »

Tableau 1-2:Description textuelle du cas d'utilisation «Inscription»

Tableau 1-3:Description textuelle du cas d'utilisation «Consulter annonces »

Tableau 1-4:Description textuelle du cas d'utilisation «poster travail »

Tableau 1-5:Description textuelle du cas d'utilisation«Consulter devoir »

Tableau 1-6:Description textuelle du cas d'utilisation «traitement de demande »

Tableau 1-7:Description textuelle du cas d'utilisation«Telecharger document »

Tableau 1-8:Description textuelle du cas d'utilisation «demande d'accès »

Tableau 1-9:Description textuelle du cas d'utilisation «gestion des classes »

Tableau 1-10:Description textuelle du cas d'utilisation «gestion des devoirs »

Tableau 1-11:Description textuelle du cas d'utilisation «Gestion des annonces »

Tableau 1-12:Description textuelle du cas d'utilisation« gestion d'assiduité»

# **Introduction Générale**



**Aujourd’hui l’internet devient une source majeure en termes de communication et de services, Chaque école doit être en mesure de répondre et participer à ces nouveaux enjeux. Le Web est devenu un outil média important dans le monde et un outil d’usage indispensable dans le public scolaire (étudiants et enseignants) , c'est dans ce contexte se situe notre travail qui consiste à concevoir et implémenter un nouveau service en ligne, il s'agit d'une application web de gestion des classes qui permet aux différents enseignants une gestion simple et efficace des classes qu'ils dirigent et même pour les étudiants une utilisation simple et un accès facile pour pouvoir accéder aux différentes fonctionnalités d'une classe.**

**Ce présent rapport a pour objectif de présenter d'une façon détaillée les étapes d'élaboration de ce projet. Il s'articule autour de 3 chapitres : Le premier chapitre consistera à la présentation du projet, le contexte et les contraintes rencontrées.**

**Puis, le deuxième chapitre sera consacré à la partie modélisation en citant les différents utilisateurs, leurs besoins fonctionnels et les scénarios possibles des interactions avec l’application.**

**Ensuite nous entamerons dans le troisième chapitre la définition de l’essentiel du travail en déterminant le plan suivit au développement de l’application et les outils utilisés. et nous allons aussi mettre le point sur l’étape de réalisation et implémentation. A cet effet nous allons vous présenter quelques interfaces.**



# CHAPITRE I :

## Contexte général

*Ce chapitre introductif présentera la problématique du sujet ainsi que les démarches adoptées et les missions réalisées durant notre projet.*

# **Objectifs**



**L'objectif de notre projet est l'informatisation, la simplification la mise en place d'une solution qui facilite aux enseignants de créer des classes et les gérer et aux étudiants de créer des comptes et demander l'accès à une ou plusieurs classes.**

**Les enseignants pourront gérer facilement leurs classes par exemple ils peuvent gérer l'assiduité de leurs étudiants, publier des annonces et programmer des devoirs que les étudiants sont censés de les postés , de plus leurs faciliter le calcul et l'affichage des notes.**

**En outre l'étudiant à travers ce projet il n'aura plus à se déplacer vers la classe pour déposer son devoir il pourra le poster en ligne et il peut aussi consulter les annonces et télécharger les documents publiés par son professeur.**

## **Missions du projet**



**La mission du projet comprend les services suivants :**

- **L'étude fonctionnelle**
- **L'étude technique**
- **La conception**
- **La réalisation**
- **Le test**
- **Le déploiement**

# Probématique



**Tout projet se heurte en amont à des difficultés qu'on doit les prendre en compte pour pouvoir espérer se développer .Il est donc capital de bien repérer ces contraintes et de déterminer comment elles pourraient être atténuées.**

**L'ensemble des contraintes affrontées peut être subdivisé en 3 catégories :**

## **1- Contraintes ergonomiques :**

**Se présentant en l'adaptation des fonctionnalités à l'utilisateur en lui permettant un accès rapide à l'information et la simplicité de l'interface pour qu'il soit compréhensible.**

## **2- Contraintes techniques :**

**Dans un premier temps, on doit assurer la maintenance et la lisibilité du code pour faciliter toute opération de modification ou d'optimisation.**

**En second temps, pour pouvoir envoyer des messages aux utilisateurs de l'application contenant leurs login et mots de passe lors de leurs inscriptions on doit être connecter à internet.**

## **3- Contraintes personnelles :**

**Se réduit dans le temps de réalisation qui ne semble pas suffisant pour donner un projet complet et professionnel.**

**En plus, il faut être sûr d'avoir les fonds propres suffisants en cas de difficulté au lancement de l'activité, vu que le langage de programmation utilisé semble compliqué puisqu'on n'était pas bien en familier.**

# Livrables

**Un livrable est le résultat du découpage d'un projet en résultats intermédiaires qui concourent à la réalisation du résultat attendu par le projet. Les principales livrables de notre application sont :**

Phase	Livrable	Date de Livraison
Analyse et Conception	Dossier de spécifications fonctionnelles	09/04/2022
Installation de la solution	Installation & code source	12/04/2022
Tests	Implémentation des interfaces	25/05/2022
Documentation	Rapport Projet	06/06/2022

Tableau 1-Livrable

# Risques

Les risques	Le type	Impact	Probabilité	Action correctives
<b>Cahier des charges incomplet</b>	Risque non bloquant	Créé une ambiguïté ce qui pourra générer un retard qui influencera probablement la date de livraison	Moyenne	-Prévoir des réunions et des points de validation avec l'encadrant au fur et à mesure de l'avancement du projet.
<b>Le temps est court</b>	Risque non bloquant	Ce qui pourra générer un retard qui influencera probablement sur la date de livraison.	Forte	-Doubler les efforts et consacrer plus de temps au projet. -Travailler en mode synergie
<b>Les pannes inattendues du Matériel</b>	Risque bloquant	Ralentissement des travaux	Forte	- Faire des autoformations -Utiliser les autres matériaux disponibles. -Recours à une réparation rapide. - Doubler l'effort et travailler un temps extra.

Tableau 2-  
Risques

# Méthodologie de travail

Le modèle du cycle en V est un modèle conceptuel de gestion de projet. Il permet, en cas d'anomalie, de limiter un retour aux étapes précédentes.

Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés, afin d'améliorer le logiciel.

La caractéristique principale de ce processus est que chaque étape de conception fonctionne en binôme avec une phase de test (validation).

Ensuite, plus on avance dans l'étude, plus le niveau de détails est important. Par ailleurs, à partir du codage, plus on avance, les tests se déroulent dans la globalité du projet (moins de détails).

Soulignons que le cycle en V est devenu un standard dans l'Industrie logicielle depuis les années 1980, notamment l'apparition de l'Ingénierie des Systèmes. Il faut aussi souligner qu'il s'agit d'un standard conceptuel dans tous les domaines de l'Industrie.

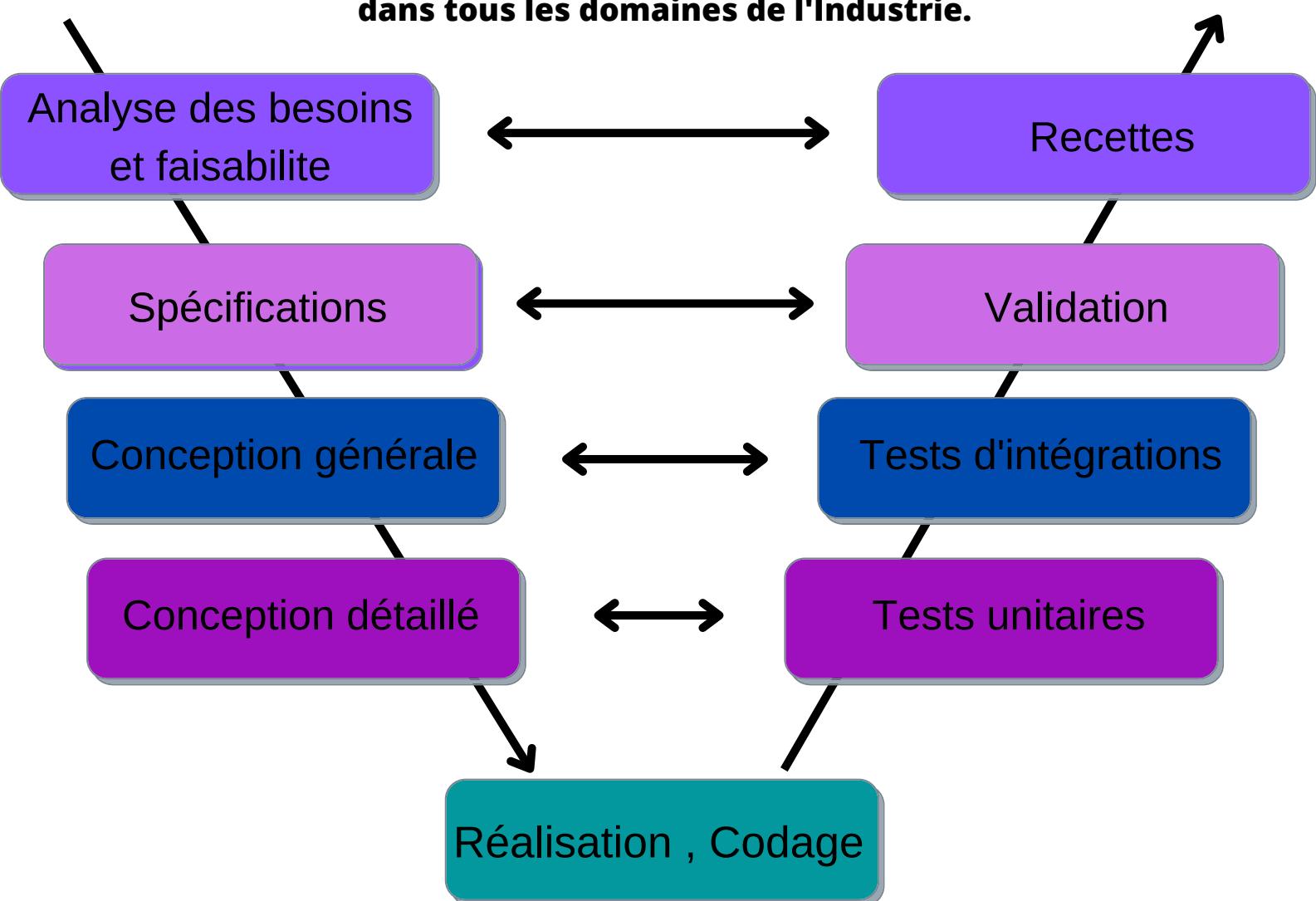


Figure 1:Cycle en V

# Cycle de vie

**Pour qu'il soit géré dans un contexte de qualité, un projet doit suivre différentes phases au terme desquelles des points de contrôle doivent être définis. Chaque étape fait l'objet d'un livrable et d'une validation à partir d'un document spécifique.**

**Cela permet de maîtriser la conformité des livrables à la définition des besoins ainsi que de s'assurer de l'adéquation aux objectifs et au délai.**

**L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement.**

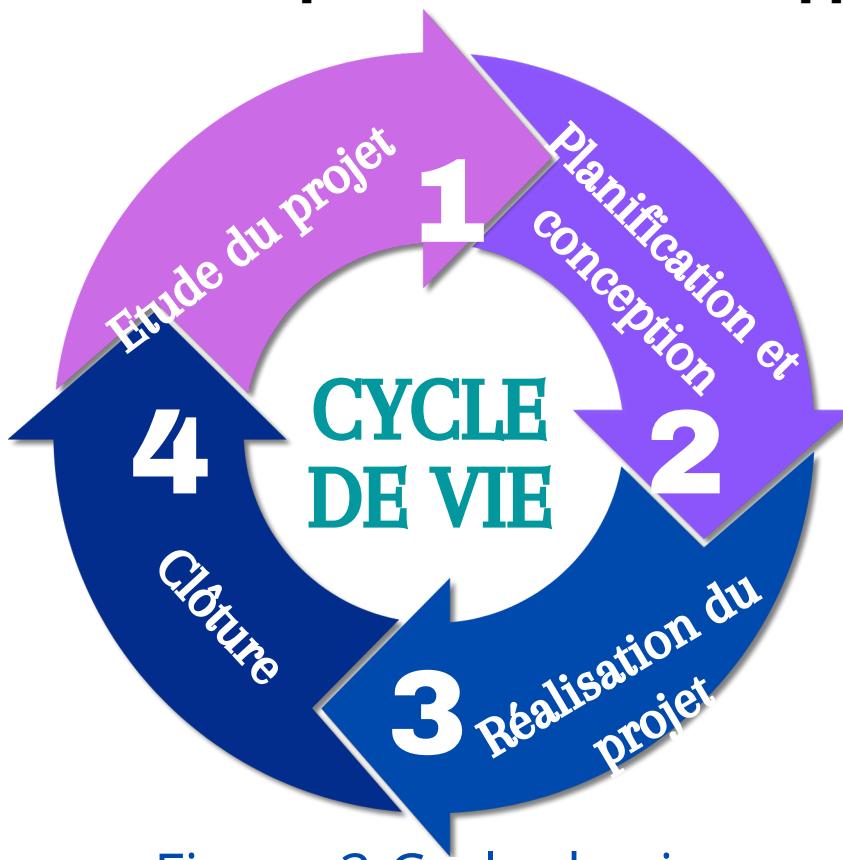


Figure 2:Cycle de vie



## • Etape d'étude du projet

**Cette première phase d'étude et d'analyse permet d'étudier et analyser les finalités du projet. Nous avons également pensé aux solutions à apporter à travers la réalisation de notre projet. Pour nous, le but était avant tout d'identifier la meilleure façon de réaliser le projet.**

## • Etape de planification et conception

**Processus qui consiste à représenter, et maquetter les diverses fonctions du système d'une manière qui permettra d'obtenir rapidement un ou plusieurs programmes réalisant ces fonctions. Elaboration des spécifications de l'architecture générale du logiciel.**

## • Etape de réalisation du projet

**Cette étape représente le corps du projet. C'est aussi l'étape qui consomme le plus de temps et d'énergie puisqu'il s'agit de mettre en œuvre concrètement les éléments planifiés. Dans cette étape, nous avons débuté par la modélisation du projet en cherchant les outils de travail, puis passer à la phase de l'implémentation.**

## • Etape du clôture

**Après avoir franchi toutes les étapes de la réalisation du projet, il s'agit dans cette phase de faire la présentation générale de notre travail en expliquant tous les livrables.**

# Planning prévisionnel du projet

**Le diagramme de GANTT est un outil de planification des tâches nécessaires pour la réalisation d'un projet qui consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de réalisation.**

**Le diagramme de GANTT suivant vous montre l'enchaînement des tâches de notre projet sous forme de tâches et leurs durées.**

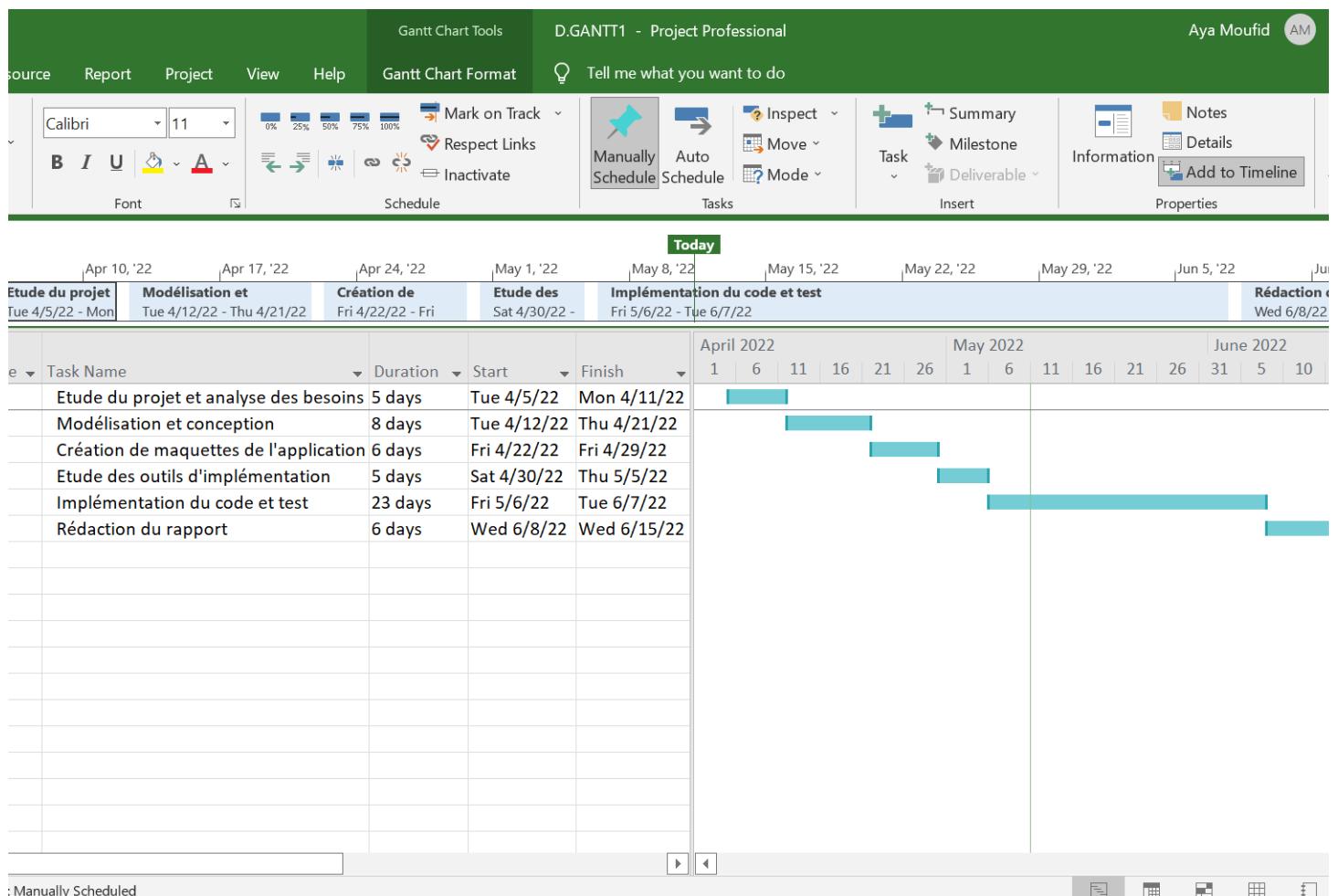


Figure3 : Diagramme de GANTT prévisionnel

# Planning réel du projet

**Finalelement, le diagramme de GANTT initial n'a pas tout à fait été respecté, beaucoup de nouvelles tâches sont apparues et certaines tâches nous ont pris plus de temps que prévu.**

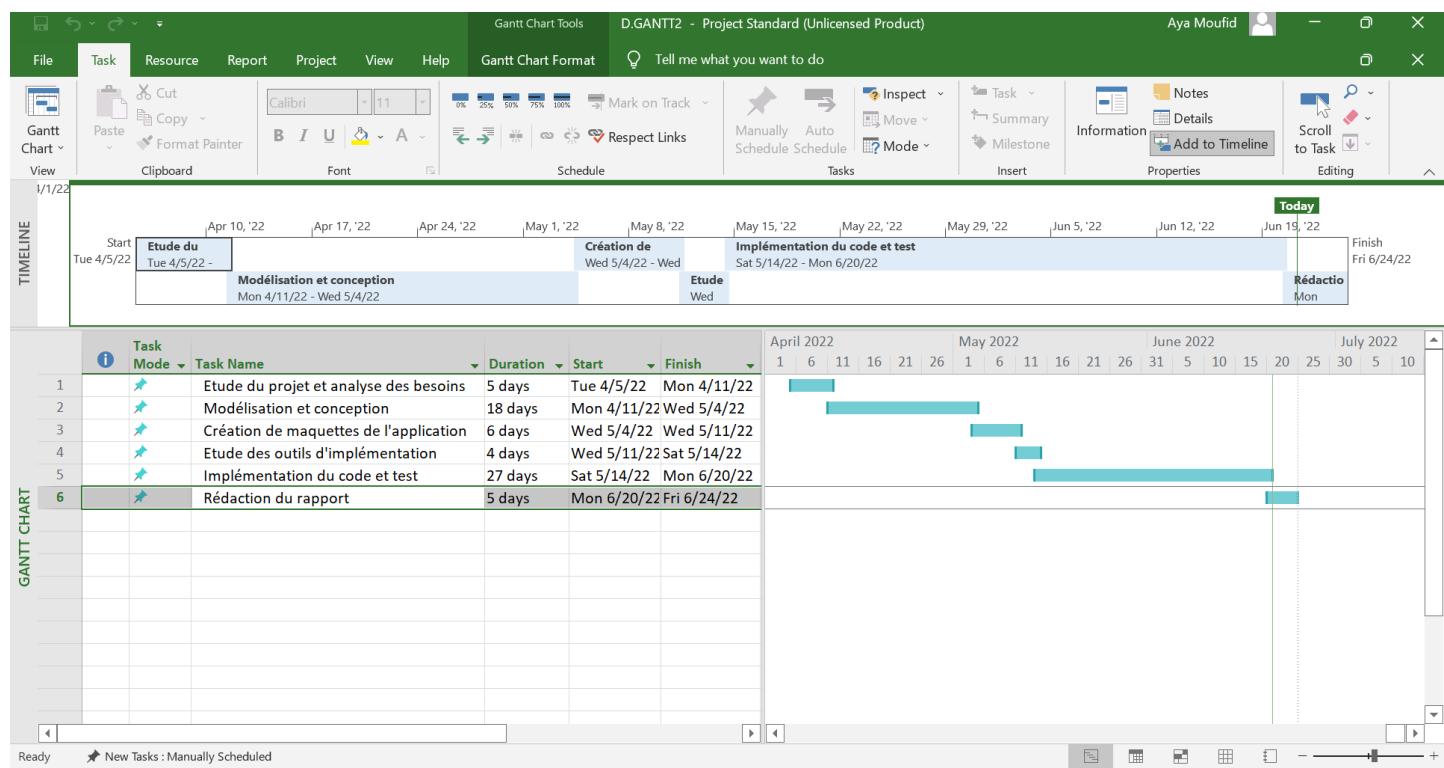


Figure4 :Diagramme de GANTT réel

# Conclusion



**Dans le présent chapitre, nous avons présenté les objectifs et la problématique de notre projet , ainsi que la démarche avec une planification du temps prévisionnel et une autre du temps réel, ce qui nous permet de suivre tout au long du projet pour assurer une bonne mise en œuvre de l'application.**

**Le chapitre suivant sera consacré à l'analyse et la conception du projet .**



# **CHAPITRE II :**

## **Analyse et conception**

**Dans ce chapitre nous présentons les méthodologies de conception que nous avons utilisée dans la conception de notre projet. Donc nous présentons une spécification et une analyse des besoins fonctionnels et non fonctionnels de l'application. Nous enchainons alors avec l'architecture de notre application et par suite nous présentons les différents diagrammes, à savoir les, de cas d'utilisation, de séquence , de classe, et aussi le modèle conceptuel de données afin de spécifier de façon détaillée les aspects fonctionnels, dynamiques et statiques du système. Nous présentons aussi une maquette générale qui représente les différents pages de l'application.**

# Besoins fonctionnels



## 1- Authentification

**Cette partie de l'application est une partie très importante. Si l'utilisateur est déjà inscrit cela permet son authentification. Pour les nouveaux, elle représente une page pour s'inscrire.**

## 2- Gestion des classes

**Cette fonction permet à l'enseignant de créer une classe composée de plusieurs étudiants, documents, devoirs, annonces, séances. Après son authentification, l'enseignant peut choisir à créer des classes. Pour ce faire, il doit remplir les informations de sa classe(Nom classe, Nom formation, Semestre, Année universitaire).**

**Après avoir entré toutes les informations relative à une classe,l'enseignant est invité à accepter les demandes des étudiants.**

## 3- Gestion des documents

**Après avoir affecté les étudiants à une classe, l'enseignant doit renseigner à l'application des documents qui seront téléchargés par l'étudiant en précisant les informations d'un document(Titre, Description, Date de mise en ligne, Catégorie), et c'est à l'enseignant de choisir la catégorie du document.**



## 4- Gestion des devoirs

**Après la création de la classe, l'enseignant à le choix de publier des devoirs en insérant les informations d'un devoir(Titre, Énoncé, Dernier délai de dépôt, Format demandé). Une fois le devoir est posté, un email contenant toutes les informations du devoir est automatiquement envoyé aux étudiants par l'application. Après la réception du mail, l'étudiant a le droit de poster son travail avant le Deadline.**

## 5- Gestion des annonces

**Après la création de la classe, l'enseignant peut choisir de publier des annonces en insérant les informations nécessaires(Nom, Date de publication, Description). Juste après la publication d'une annonce, l'application envoie à chaque étudiant un mail contenant toutes les informations. L'étudiant peut choisir à consulter les annonces via l'application.**

## 6- Gestion des notes

**En s'assurant que toutes les notes des devoirs, des contrôles, d'assiduité, des examens des 2 sessions sont générées, l'enseignant responsable peut choisir les pourcentages de chaque note pour le calcul de la note générale. Juste après le calcul, 3 fichiers Excel sont générés.**

# Besoins non fonctionnels



**Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement.**

- **Rapidité** : L'application doit optimiser les traitements pour avoir un temps de génération de schéma raisonnable.
  - **Efficacité** : L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.
- **La manipulation** : L'application doit avoir une manipulation simple facilitant son utilisation aussi bien pour les enseignants que pour les étudiants.
- **Performance** : décrivent les performances d'exécution du système, généralement en termes de temps de réponse. - Ex :  
**(application Web) Temps de chargement d'une page** : Le chargement d'une page Web dans le navigateur ne devrait pas prendre plus de 15 secondes en condition normale.
  - **La sécurité** : il est important de sécuriser l'accès à L'application ,raison pour laquelle un mot de passe est nécessaire lors de l'authentification.
- **La disponibilité** : Notre application doit être disponible à tout instant pour être utilisée par n'importe quel utilisateur, et doit être facilement accessible via n'importe quel ordinateur.



# CONCEPTION

**Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. Les différents types des diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.**

**Dans cette partie nous allons présenter les diagrammes figurant notre application.**

# Diagramme de cas d'utilisation

## 1- Definition

**Les cas d'utilisation permettent de décrire le système d'un point de vue utilisateur, ils modélisent les actions/réactions du système avec les acteurs. Dans notre projet, nous n'avons que deux acteurs qui sont l'enseignant et l'étudiant.**

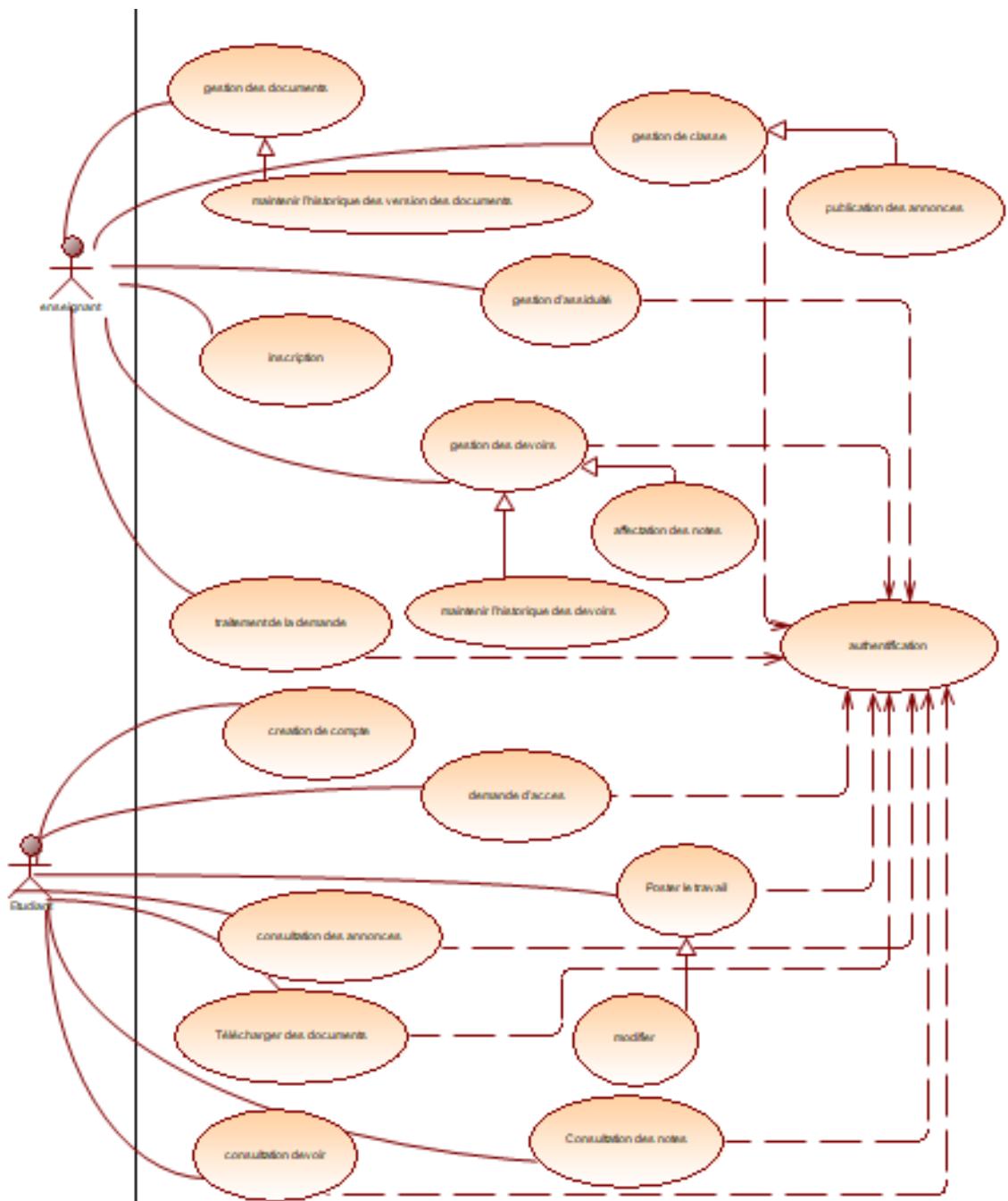


Figure 5: Diagramme de cas d'utilisation



## 2- Description textuelle

<b>Acteurs concerné</b>	<b>Etudiant , Enseignant</b>
<b>Pré -condition</b>	Compte existant
<b>Post-condition</b>	Autorisation d'accès accordée (L'utilisateur doit être créé dans la base de données et connaître ses identifiant)
<b>Scénario nominal</b>	1-Le système affiche le formulaire d'identification 2-L' étudiant saisi le login et le mot de passe 3-L' étudiant clique sur Login 4-Le système vérifie les données saisies 5-Compte existe Le système le renvoie vers la page d'accueil d'un l'étudiant.
<b>Scénario d'exception</b>	1-L'utilisateur n'as pas saisie les bons identifiants. 2-Le système renvoie un message d'erreur et signale à l'utilisateur de recommencer 3-L'utilisateur n'existe pas dans la base de données.

**Tableau 1-1 Description textuelle du cas d'utilisation «S'authentifier »**

<b>Acteurs concerné</b>	<b>Etudiant , Enseignant</b>
<b>Pré - condition</b>	Consulter l'application
<b>Post- condition</b>	Créer un compte dans l'application
<b>Scénario nominal</b>	<ul style="list-style-type: none"><li>• l'étudiant clique sur s'inscrire</li><li>• Le système affiche le formulaire d'inscription</li><li>• étudiant rempli les champs nécessaires et valide</li><li>• Le système vérifie les données saisies</li><li>• Le système enregistre le nouvel utilisateur et affiche un message de succès de l'opération</li><li>• Le système affiche l'espace du membre</li></ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"><li>• En cas de champs vides ou données erronées, le système affiche un message d'erreur et il y aura une reprise de l'étape précédente</li><li>• Le compte existe déjà</li></ul>

**Tableau 1-2 Description textuelle du cas d'utilisation «Inscription»**



<b>Acteurs concerné</b>	<b>Etudiant</b>
<b>Pré -condition</b>	+Succès d'authentification
<b>Post-condition</b>	+La liste des annonces est affichée sur l'écran
<b>Scénario nominal</b>	+Consulter email ou l'étudiant clique sur une annonce affichée +Le système affiche la liste des annonces
<b>Scénario d'exception</b>	+ Aucune annonce n'est publiée

**Tableau 1-3 Description textuelle du cas d'utilisation «Consulter annonces »**

<b>Acteurs concerné</b>	<b>Etudiant</b>
<b>Pré - condition</b>	+Succès d'authentification
<b>Post-condition</b>	+un nouveau travail ajouté
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. l'étudiant choisit l'ajout d'un nouveau travail</li> <li>2. le système affiche l'espace d'ajout</li> <li>3. l'étudiant saisit les informations à remplir sur le nouveau travail</li> <li>4. Le système vérifie les données</li> <li>5. Le système enregistre le travail dans la base de données.</li> <li>6. Le système affiche un message de succès</li> </ol> <p>L'étudiant peut aussi modifier son travail</p>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• Champs obligatoires non valides ou vides</li> <li>• Délai passé</li> <li>• le système affiche un message d'erreur</li> <li>• reprendre l'étape de la saisie des informations à remplir sur le nouveau travail</li> </ul>

**Tableau 1-4 Description textuelle du cas d'utilisation «poster travail»**



<b>Acteurs concerné</b>	<b>Etudiant , Enseignant</b>
<b>Pré - condition</b>	<ul style="list-style-type: none"> <li>• Succès d'authentification</li> </ul>
<b>Post- condition</b>	<ul style="list-style-type: none"> <li>• Devoir consulté.</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• Consulter email ou l'étudiant clique sur le devoir affiché</li> <li>• Le système affiche les informations du devoir</li> <li>• L'étudiant peut consulter maintenant le devoir</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• Devoir n'existe pas</li> <li>• Aucun devoir n'est publié</li> </ul>

**Tableau 1-5 Description textuelle du cas d'utilisation «Consulter devoir»**

<b>Acteurs concerné</b>	<b>Enseignant</b>
<b>Pré -condition</b>	<ul style="list-style-type: none"> <li>• Succès d'authentification.</li> <li>• La demande existe déjà</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Demande traité</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• l'enseignant demande l'affichage de la liste des demandes</li> <li>• Le système affiche l'ensemble des demandes</li> <li>• l'enseignant choisit la demande à traité</li> <li>• l'enseignant vérifie les données de l'étudiant et appuie sur valider</li> <li>• le système affiche un message de confirmation</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• l'étudiant annule sa demande</li> </ul>

**Tableau 1-6 Description textuelle du cas d'utilisation «traitement de demande»**



<b>Acteurs concerné</b>	<b>Etudiant</b>
<b>Pré -condition</b>	<ul style="list-style-type: none"> <li>• Succès d'authentification.</li> <li>• L'étudiant visualise les informations du document sélectionné</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Document est téléchargé</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• L'étudiant clique sur le bouton télécharger document</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• Aucun document n'est posté</li> </ul>

Tableau 1-7 Description textuelle du cas d'utilisation «Telecharger document»

<b>Acteurs concerné</b>	<b>Etudiant</b>
<b>Pré -condition</b>	<ul style="list-style-type: none"> <li>• Succès d'authentification.</li> <li>-</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• Demande envoyé</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• Le système affiche l'ensemble des classes</li> <li>• l'étudiant choisit une classe</li> <li>• l'étudiant clique sur le bouton <u>demande d'accès</u></li> <li>• L'étudiant remplit les champs nécessaires dans le formulaire <u>demande d'accès</u></li> <li>• Le système vérifie les données saisies</li> <li>• Le système envoie une demande à l'enseignant responsable</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• Champs obligatoires non valides ou vides.</li> </ul>

Tableau 1-8 Description textuelle du cas d'utilisation «demande d'accès»



<b>Acteurs concerné</b>	<b>ENSEIGNANT</b>
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>Succès d'authentification.</li> </ul>
<b>Post condition</b>	<ul style="list-style-type: none"> <li>Classe ajouté – Classe supprimé – Classe modifié</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>Le système affiche la page des classes</li> <li>L'enseignant clique sur l'action désirée (Ajout-Suppression-Modification)</li> <li>L'enseignant complète le traitement demandé</li> <li>Le système vérifie les données saisies</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>Classe existe déjà</li> <li>Champs non valides ou vides.</li> </ul>

Tableau 1-9 Description textuelle du cas d'utilisation «gestion des classes»

<b>Acteurs concerné</b>	<b>ENSEIGNANT</b>
<b>Pré -condition</b>	<ul style="list-style-type: none"> <li>Succès d'authentification.</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>Devoir publié – Devoir supprimé – Devoir modifié – Notes affectées – historique maintenu</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>Le système affiche la page des devoirs</li> <li>L'enseignant clique sur le bouton désiré (Ajout-Suppression-Mise à jour-Affecter les notes-Gérer l'historique)</li> <li>L'enseignant complète le traitement demandé</li> <li>Le système vérifie les données saisies</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>Classe existe déjà</li> <li>Champs non valides ou vides.</li> </ul>

Tableau 1-10 Description textuelle du cas d'utilisation «gestion des devoirs»



<b>Acteurs</b>	<b>ENSEIGNANT</b>
<b>Pré Condition</b>	Succès d'authentification
<b>Post Condition</b>	Annonce publié – Annonce supprimé – Annonce modifié
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• Le système affiche la page des annonces</li> <li>• L'enseignant choisit soit de publier supprimer ou modifier une annonce</li> <li>• Le système affiche un message de confirmation</li> <li>• Le système enregistre les données saisies</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• Annonce existe déjà</li> <li>• Champs non valides ou vides.</li> </ul>

**Tableau 1-11 Description textuelle du cas d'utilisation  
«Gestion des annonces»**

<b>Acteurs concerné</b>	<b>Etudiant</b>
<b>Pré -condition</b>	<ul style="list-style-type: none"> <li>• Succès d'authentification.</li> <li>• Fin de séance</li> </ul>
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• contrôler l'assiduité</li> </ul>
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>• l'enseignant sélectionne la classe</li> <li>• l'enseignant enregistre les absences constatées des étudiants</li> </ul>
<b>Scénario d'exception</b>	<ul style="list-style-type: none"> <li>• pas d'exception</li> </ul>

**Tableau 1- 12 Description textuelle du cas d'utilisation «gestion d'assiduité»**

# Diagramme de séquence

## 1-Définition

**Un diagramme de séquence est un objet qui définit le comportement dynamique du système. Il modélise les interactions entre les objets du système en respectant l'ordre chronologique. Il traduit la communication entre ces différents objets par des appels et des méthodes.**



## 2-Diagramme de séquence : Authentification

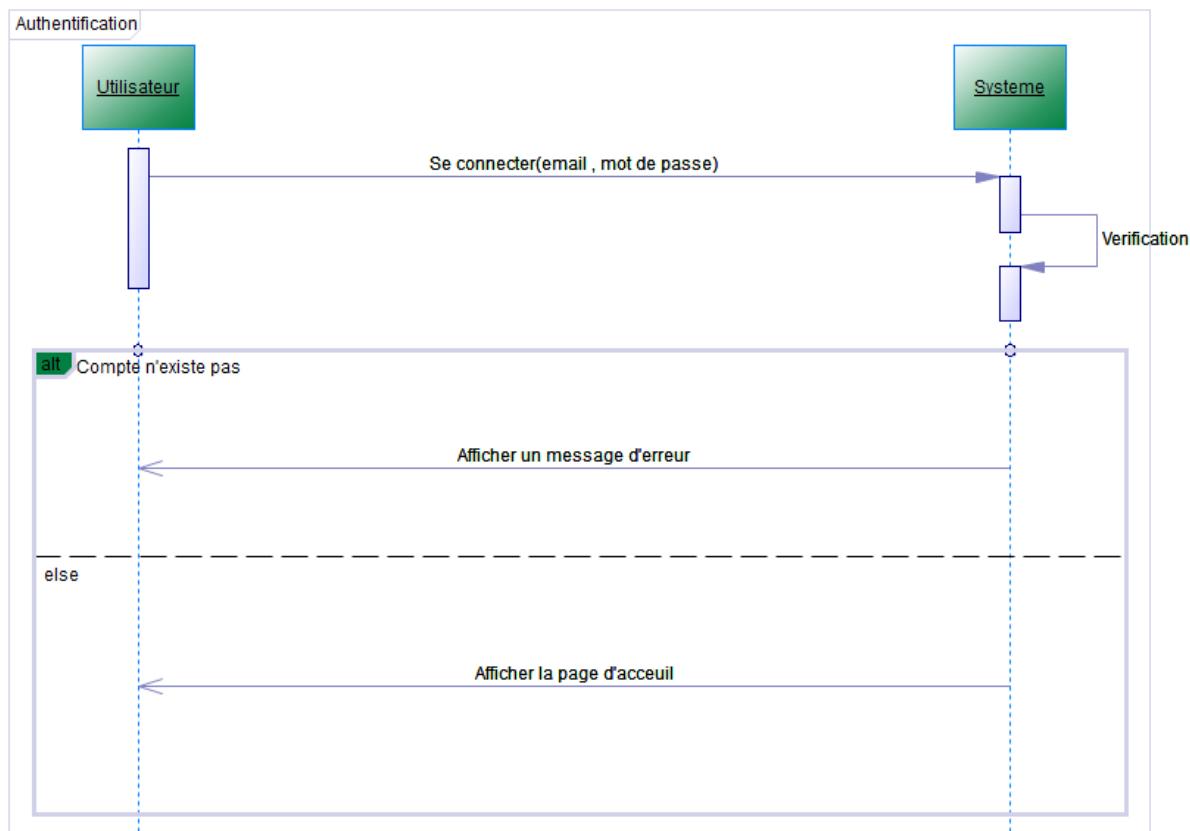


Figure 6: Diagramme de séquence Authentification

## Description

**Ce scénario est déclenché par l'enseignant ou par l'étudiant, en premier lieu un formulaire s'affiche contenant l'email et le mot de passe que l'utilisateur doit remplir. Dès que l'utilisateur saisit ses informations. Une vérification se lance dans la base de données, si les données sont valides, le système permet à l'utilisateur l'accès, sinon, un message de refus lui est adressé l'incitant à ressaisir ses coordonnées.**

### 3-Diagramme de séquence : Traitement de demande d'accès

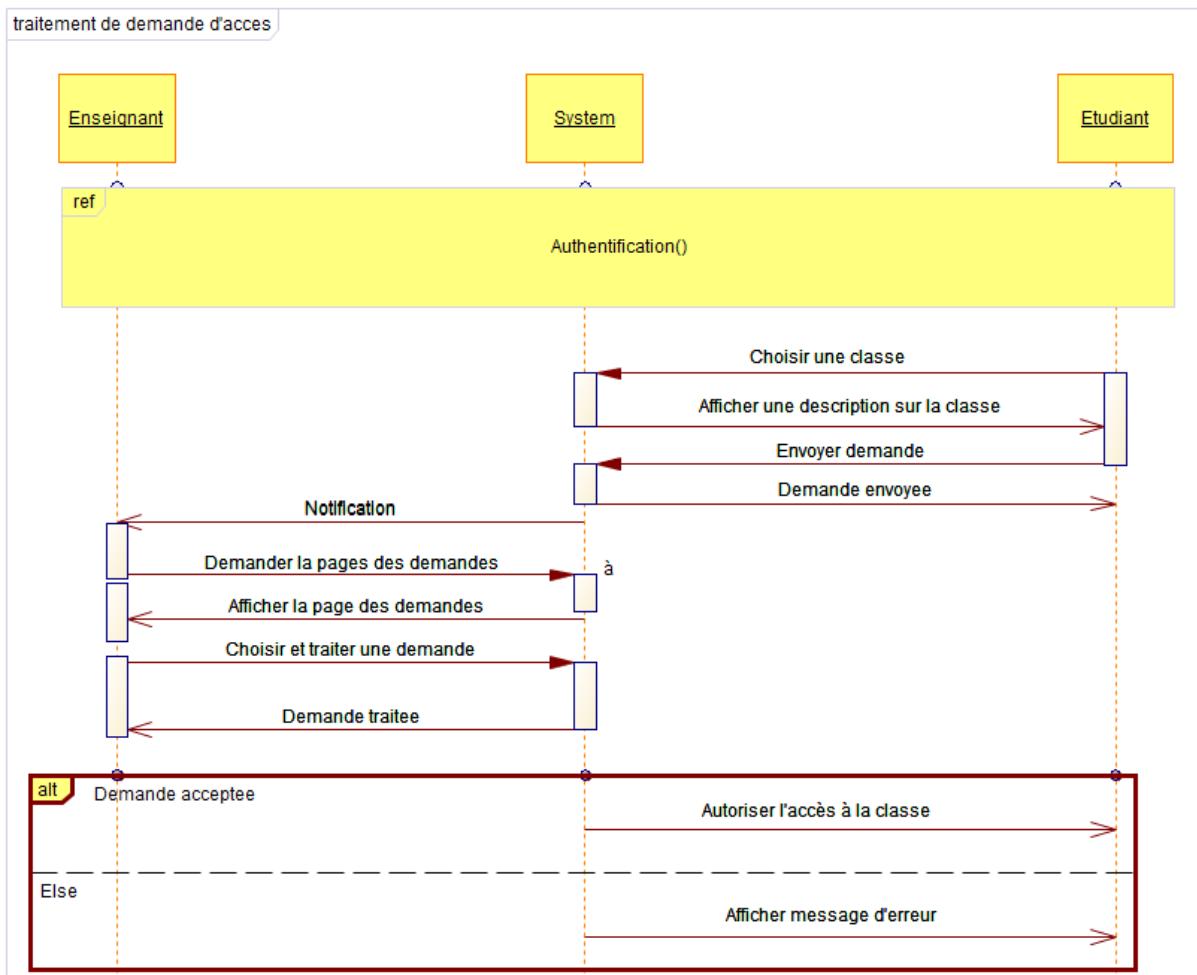


Figure7-Diagramme de séquence :Traitement de demande d'accès

### Description

**Diagramme de séquence qui représente le cas d'utilisation « Traitement de demande d'accès » qui est une action faite par un étudiant après authentification dans laquelle il a le droit de faire une demande d'accès à une classe précise en envoyant une demande qui sera traité par l'enseignant.**



## 4-Diagramme de séquence : Gestion des devoirs

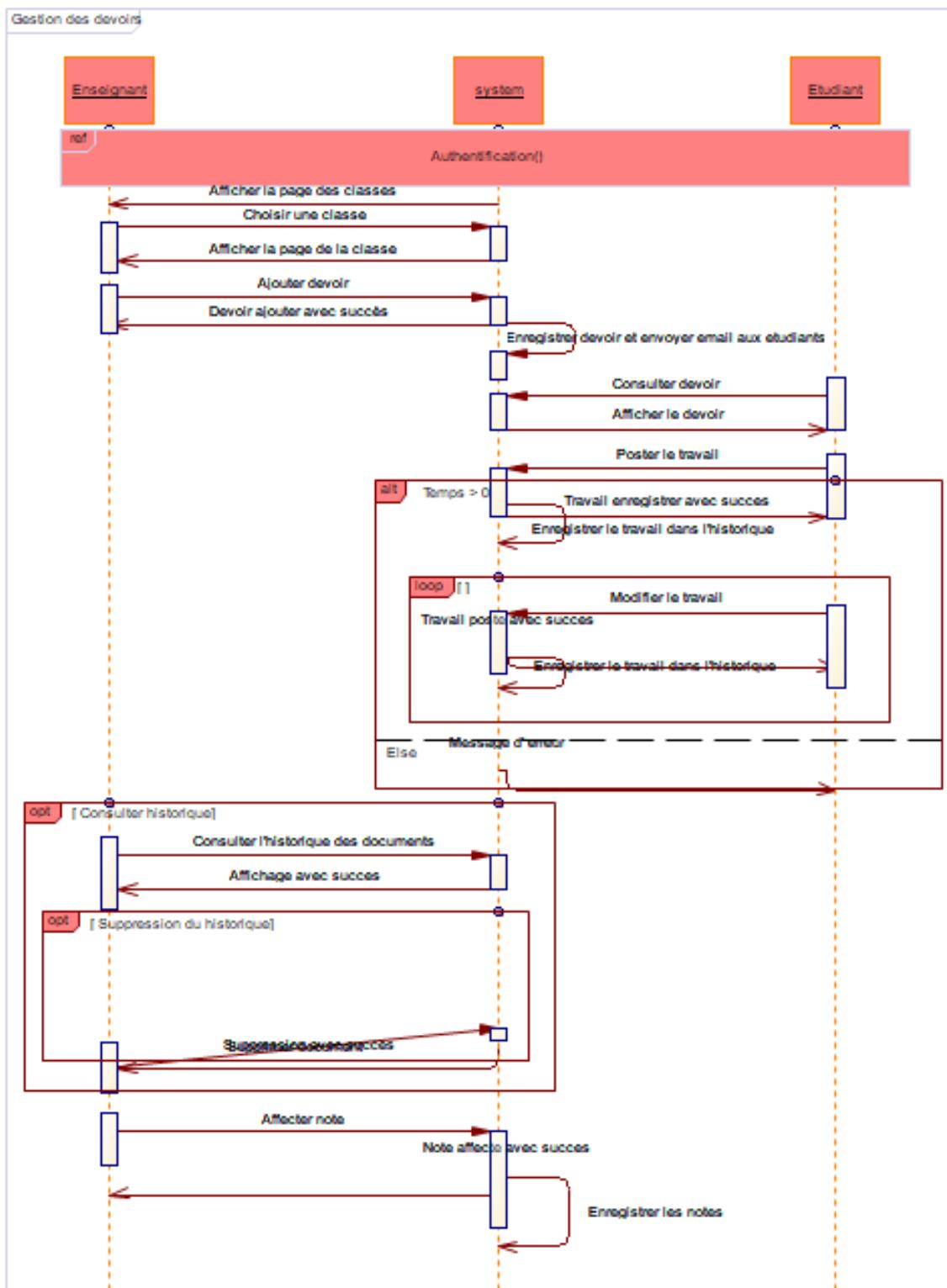


Figure 8 : Diagramme de séquence Gestion des devoirs



## Description

**D'après une authentification faite par l'enseignant ou l'étudiant. Le système affiche les classes appartenant à l'enseignant, et ce dernier choisit une de ces classes et le système l'affiche. Après il ajoute un devoir et le système l'enregistre, publie et envoie un email aux étudiants.**

**L'étudiant peut consulter le devoir et le poster dans les délais fixés. Tant que le délai est non terminé il peut poster son travail plusieurs fois et à chaque fois le système l'enregistre.**

**L'enseignant a aussi le droit de consulter l'historique des documents postés par les étudiants et il peut supprimer les anciennes versions postées. Dès que le délai est terminé l'enseignant affecte des notes et le système les enregistrent.**

# Diagramme de Classe

## 1-Définition

**Le diagramme de classe est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre elles.**

## 2-Diagramme de classe

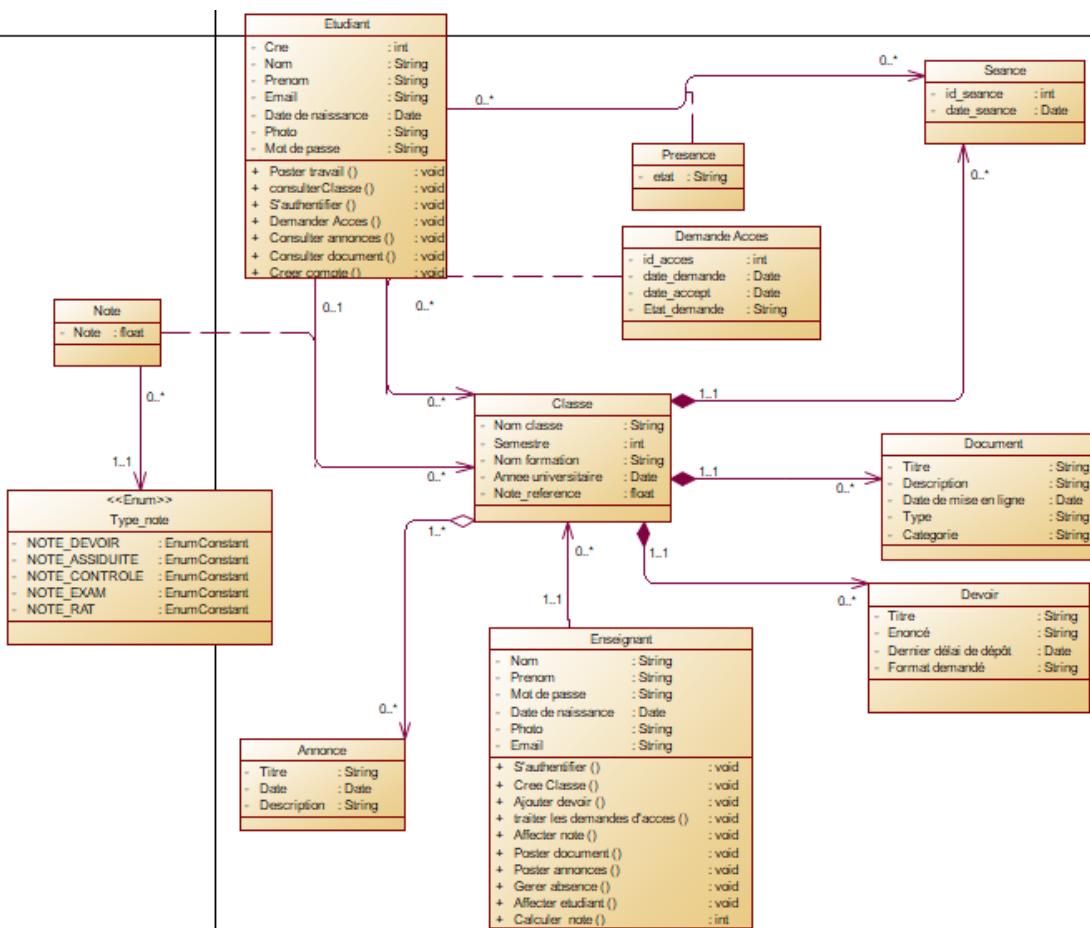


Figure9 : Diagramme de classe



## Description

**Ce diagramme décrit la structure de notre application en modélisant les entités en termes de classes et les relations entre ces classes.**

**Les classes qui nous seraient utiles pour ce projet sont :**

**La classe étudiant** qui a comme attributs nom, prénom, cne, email, date de naissance, photo et mot de passe. Les méthodes présentées dans cette classe sont comme suit : la méthode S'authentifier pour que l'étudiant s'authentifie, Demander Accès () pour que les étudiants pourront accéder à une classe, Consulter annonces pour que l'étudiant puisse consulter les annonces postées par son enseignant et consulter Classe () pour consulter toutes les classes auxquelles il est déjà inscrit, Poster travail () pour poster les devoirs programmés par l'enseignant et Consulter document () pour consulter les documents postés par l'enseignant de la classe.

**La classe classe** comporte les attributs suivants nom de classe de type string, semestre, nom formation et année universitaire de type date. Cette classe n'a pas de méthodes.

**La classe association** demande d'accès associé à la classe étudiant et classe pour que chaque étudiant passe par une demande d'accès pour accéder à une classe qui sera traitée par un enseignant.

**La classe enseignant** qui a un nom, prénom, mot de passe, date de naissance, photo et email. Les méthodes présentées dans cette classe sont comme suit : la méthode S'authentifier pour que l'enseignant s'authentifie, Cree Classe() pour la création des classes, Ajouter devoir() pour programmer des devoirs pour une classe, traiter les demandes d'accès() pour traiter la demande de chaque étudiant, Affecter note() pour affecter une note pour chaque devoir rendu, Poster document() pour associer un ou plusieurs documents à une classe, Poster annonces() pour publier des annonces textuelles pour une ou plusieurs classes à la fois.



**Gérer absence() pour gérer l'assiduité des étudiants d'une classe par séance et Affecter étudiant() pour affecter un étudiant à une classe précise.**

**La classe document** qui a comme attributs titre de type **string**, **description**, **date de mise en ligne** et le **type du document**.

**La classe devoir** qui a comme attributs titre ,énoncé , dernier délai de dépôt (Date & Heure) , format demandé (Extensions possibles).

**La classe annonce** qui a comme attributs nom et date.

**Et La classe séance** qui a comme attributs id\_séance, état\_assiduité , note\_assiduité et note\_référence.

On a utilisé le concept de **composition** car un devoir ,une séance ou un document concernent une seule classe. Et on a utilisé aussi le concept de **l'agrégation** puisque une classe se compose d'ensemble d'annonces.

Concernant les relations entre les classes, on a traité une **association** « appartient » entre étudiant et classe dont les cardinalités 0..\* et 0..\* pour expliquer qu'un étudiant peut appartenir à plusieurs classes . De la même façon on a une autre association entre enseignant et classe dont les cardinalités 1..1 et 0..\* qui reflète qu' un enseignant peut gérer 0 ou n classes et une classe peut être gérée par un seul enseignant.

# 1-Définition

**Un modèle conceptuel de données (MCD) est la représentation la plus abstraite des données qui seront utilisées par le système d'information, ces données sont représentées sous forme d'entités et d'associations**

## 2-Modèle conceptuel de données

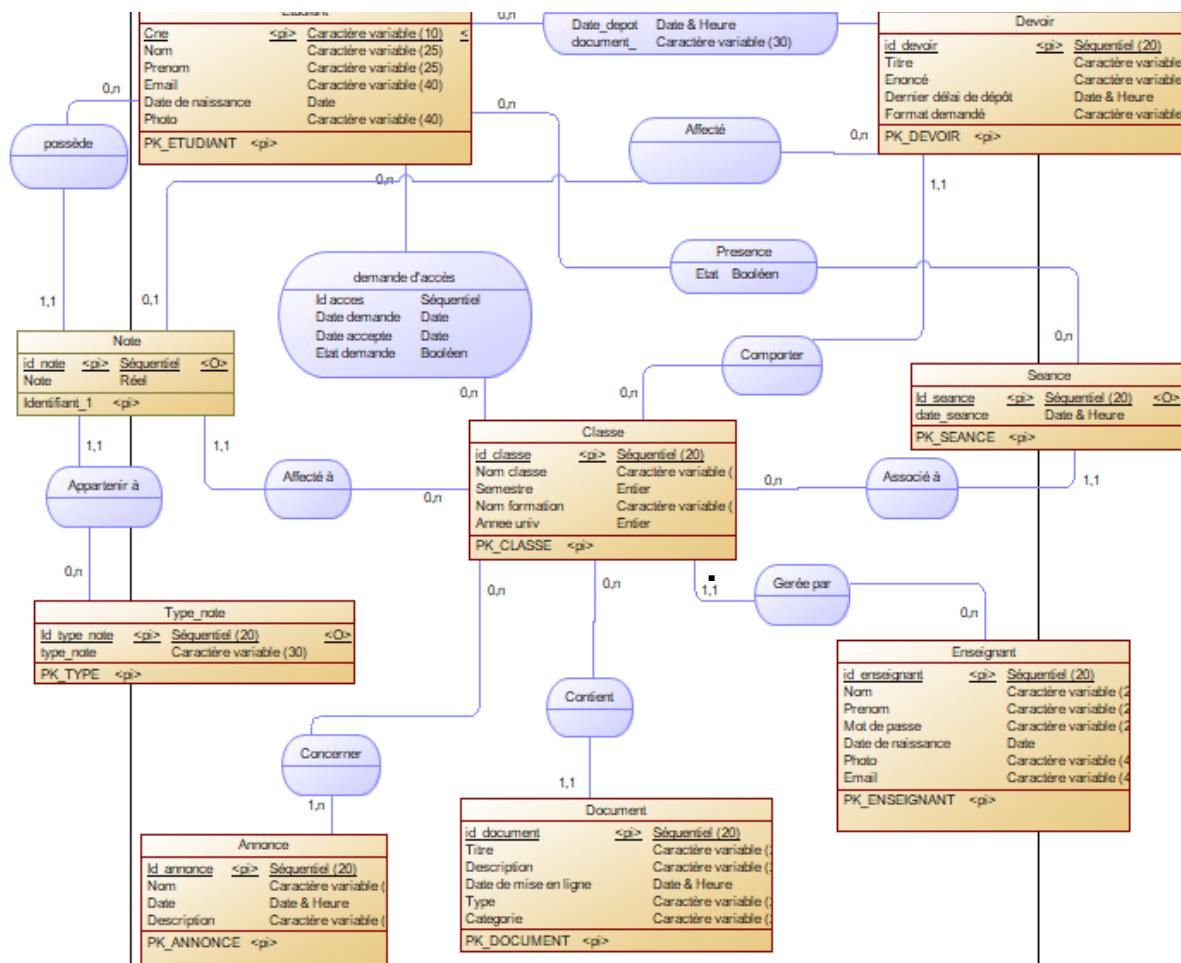


Figure 10 : Modèle conceptuel de données



## Description

**Notre premier acteur de notre application est l'enseignant .**

**Ayant comme clé primaire 'id\_enseignant', il peut créer plusieurs classes et les gérer, comme il peut ne rien créer.**

**Pour l'étudiant, il est identifiée par son 'CNE' chaque étudiant peut ne déposer aucun devoir comme il peut éventuellement déposer plusieurs, sachant qu'il y a une association reliant les deux entités qui est la classe association dépôt qui possède comme attribut la date de dépôt du devoir de l'étudiant et le travail posté par lui puisqu'elle dépend des deux à la fois. Notons à la fin qu'un étudiant peut assister à plusieurs séances.**

**Ensuite il peut demander accès à des divers classes comme il peut ne rien demandé. De même, l'association reliant entre eux contient tous les attributs dépendant des deux à la fois**

**En ce qui concerne note elle ne peut être contenue que dans une seule classe et concerne un seul étudiant.**

**Pour la classe il est identifiée par 'id classe', il peut être gérer par un seul enseignant et peut contenir plusieurs étudiants, de plus chaque classe contient plusieurs devoirs, documents et des séances qui sont spéciales pour elle ,n'oubliant pas qu' il contient des annonces et chaque annonce peut concerner plusieurs classes.**

# Structure de la base de données

**La figure ci-dessous représente les tables utilisés dans notre base des données. Ces tables représentent l'unité de persistance de notre application**

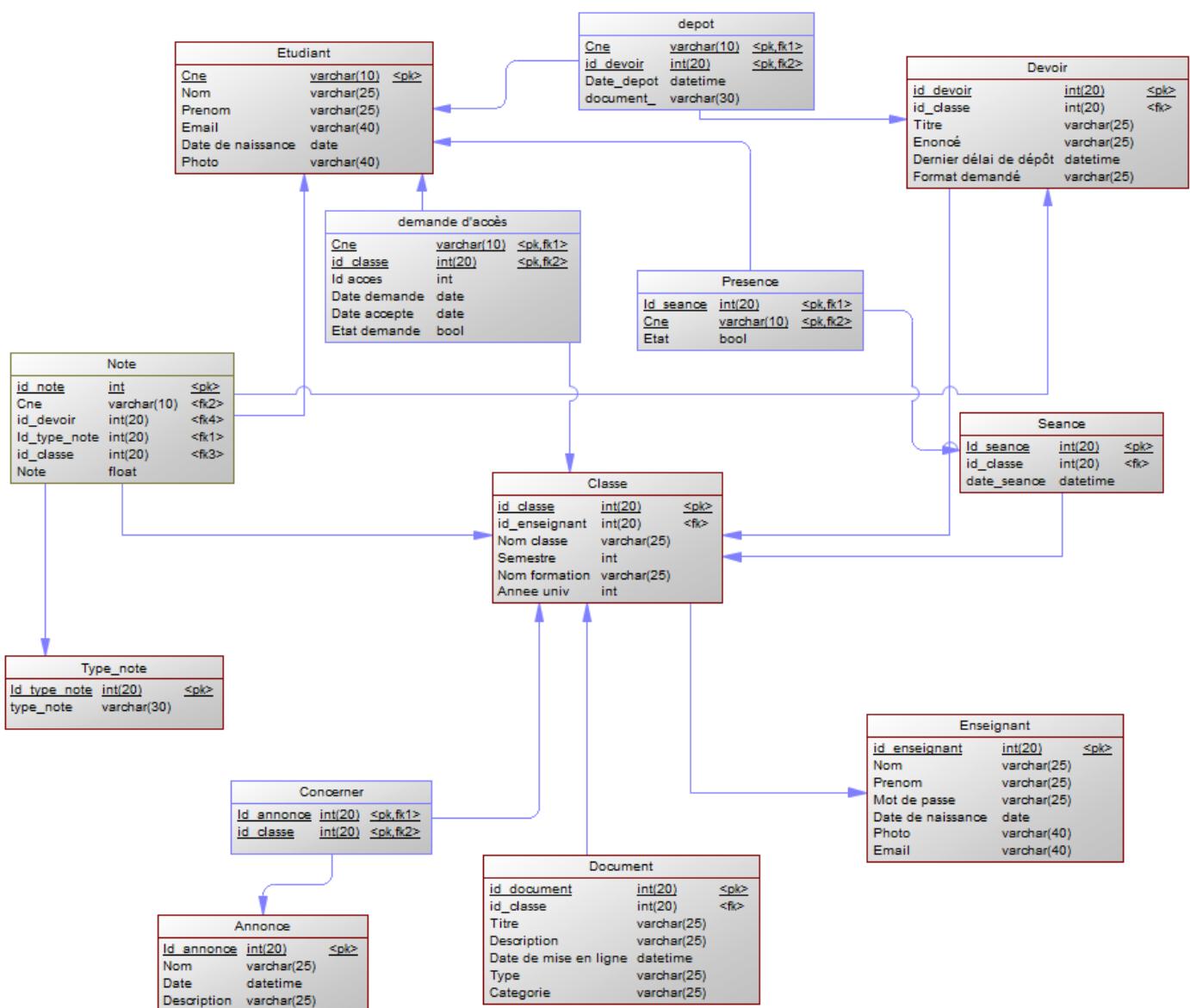
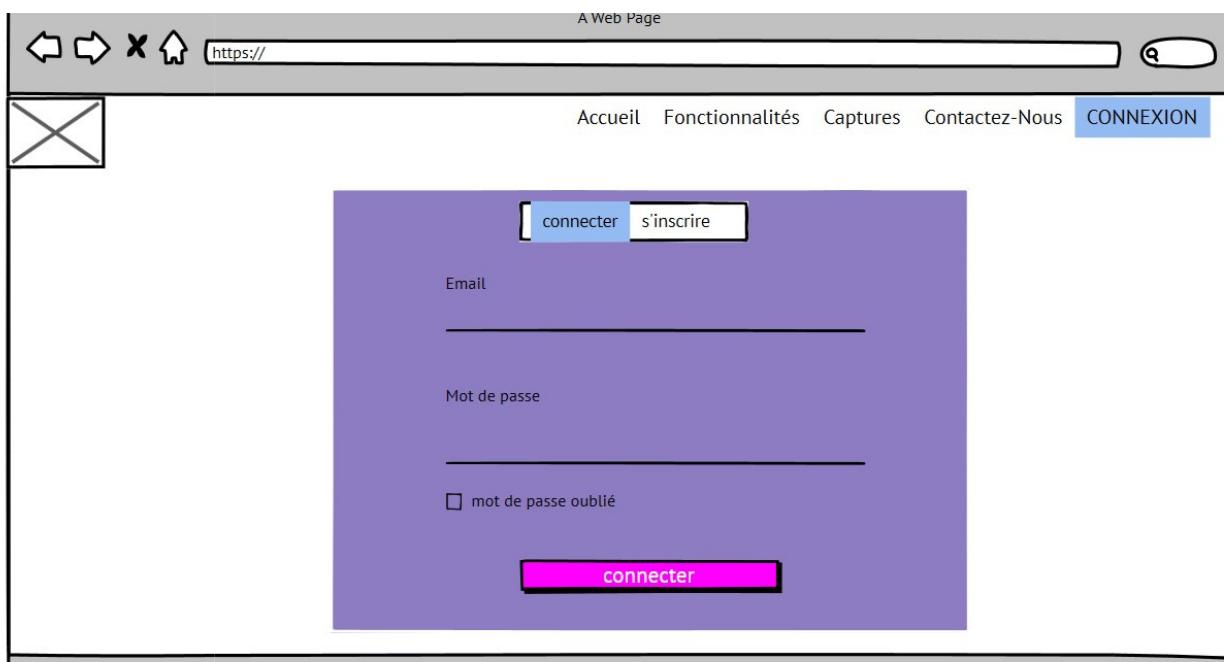


Figure 11: Modèle physique de données

# MOCKUPS

**Avant de commencer l'implémentation et la réalisation de notre application, nous avons élaboré une maquette représentative des différentes pages. Le but de cette démarche est de nous faciliter la compréhension des différentes tâches du projet, et de nous simplifier la création des pages de l'application.**

## Authentification



**Cette figure permet aux enseignants et aux étudiants de se connecter s'ils sont déjà inscrits**

Figure 12:Authentification



A Web Page

https://

The screenshot shows a web browser window titled "A Web Page". The address bar contains "https://". The main content area has a light gray background. In the top right corner, there is a blue button labeled "CONNEXION". On the left side, there is a small icon of a square with an X. The central part of the page is a purple rectangular box containing the following elements:

- A white rectangular input field with the placeholder text "Mot de passe oublié".
- An "Email" label above a horizontal line.
- An "Envoyer" button at the bottom of the purple box.

Figure 13:mot de passe oublié

**Au cas où l'utilisateur oublie son mot de passe , on lui demandera d'entrer son email et à son tour il recevra un email sur sa boite lui demandant de cliquer sur un lien lui permettant d'accéder à une nouvelle interface dans laquelle il saisira son nouveau mot de passe**

## INSCRIPTION

é

The screenshot shows a web browser window with a light gray background. At the top, there is a navigation bar with links: "Accueil", "Fonctionnalités", "Captures", "Contactez-Nous", and a blue "CONNEXION" button. On the left, there is a small icon of a square with an X. The main content area is a purple rectangular box containing the following fields:

- "connecter" and "s'inscrire" buttons (the "s'inscrire" button is highlighted).
- "Etudiant" and "Enseignant" radio buttons, with "Enseignant" selected.
- "Nom" label with an input field.
- "Prenom" label with an input field.
- "Email" label with an input field.
- "Mot de passe" label with an input field.
- "Confirmer mot de passe" label with an input field.
- "Date de naissance" label with a date input field showing "jj/mm /aaaa" and a calendar icon.
- "Photo" label with a file input field labeled "choisir un fichier" and a note "Aucun fichier n'a été sélectionné".
- An "Inscrire" button at the bottom.

Figure 14:Inscription d'un nouveau enseignant

Accueil   Fonctionnalités   Captures   Contactez-Nous   CONNEXION

Etudiant    Enseignant  
Cne

Nom \_\_\_\_\_  
Prenom \_\_\_\_\_  
Email \_\_\_\_\_  
Mot de passe \_\_\_\_\_  
Confirmer mot de passe \_\_\_\_\_

Date de naissance  
jj / mm / aaaa

Photo  
 Aucun fichier n'a été sélectionné

Figure 15:Inscription d'un nouveau étudiant  
**S'ils ne sont pas inscrits, ils doivent le faire à travers l'interface d'inscription pour posséder un compte qui leur permet de se connecter**

**L'enseignant et l'étudiant n'auront pas accès aux mêmes pages:**

A Web Page

https://

LOGO

Mes classes   Annonce   NOM & PRENOM

search

Ajouter classe

Nom de la classe Description <input type="button" value="Voir"/> <input type="button" value=""/>	Nom de la classe Description <input type="button" value="Voir"/> <input type="button" value=""/>	Nom de la classe Description <input type="button" value="Voir"/> <input type="button" value=""/>
--	--	--

Figure 16:la page principale de l'enseignant

**L'enseignant aura une page qui lui affiche des classes déjà créées ou une page sans classes avec un message 'vous n'avez aucune classe'**

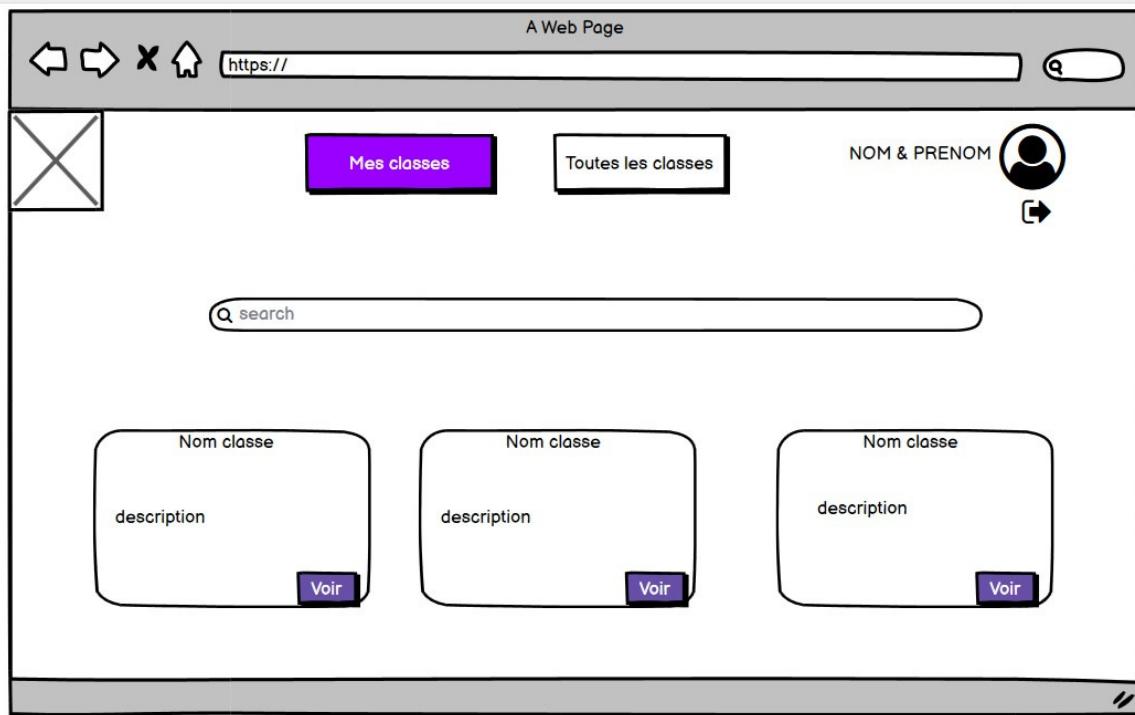


Figure 17: la page principale de l'étudiant

**L'étudiant aura une page avec des classes auxquels il appartient sinon, le système lui affiche un lien qui lui dirigera vers toutes les classes.**

## L'espace étudiant :

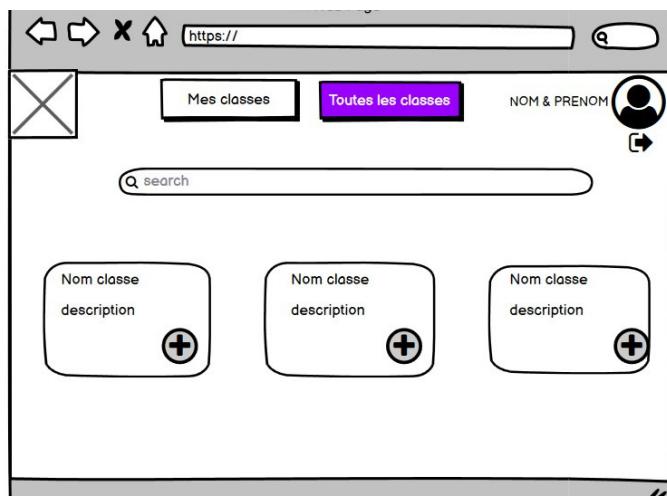


Figure 18: la pages des classes de chaque étudiant

**Cette figure affiche toutes les classes qui se trouvent dans le système .En cliquant sur l'icône +, l'étudiant envoie une demande pour avoir l'accès à la classe désirée.**

## PARTIE DES ANNONCES

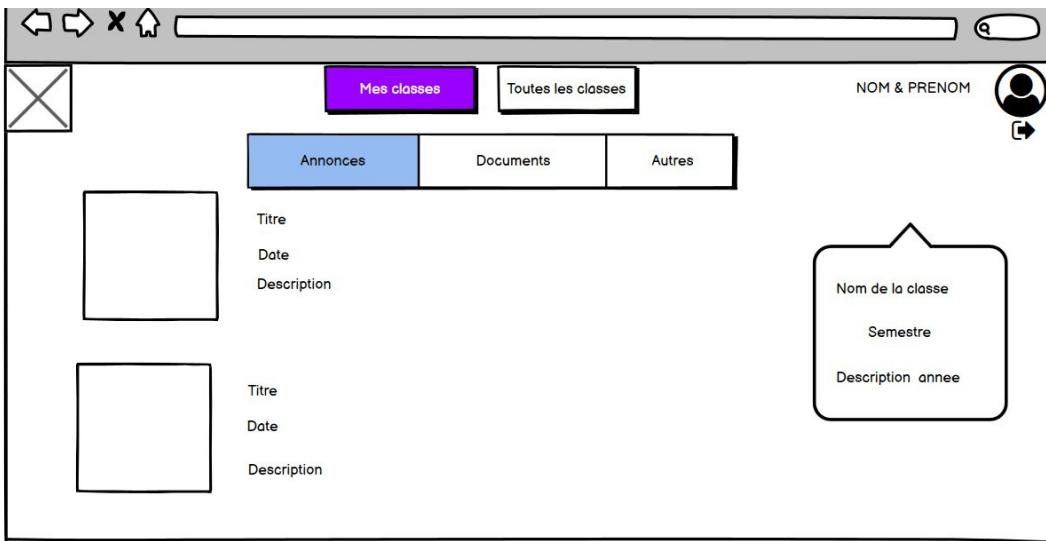


Figure 19: partie annonces

**Cette figure affiche les annonces de la classe choisie avec les informations correspondantes**

## PARTIE DES DOCUMENTS

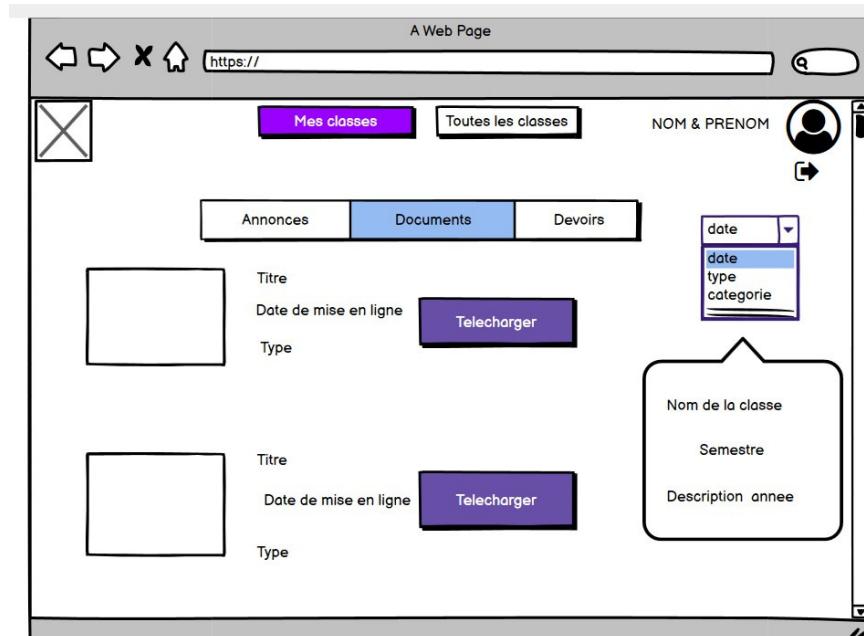


Figure 20: partie documents

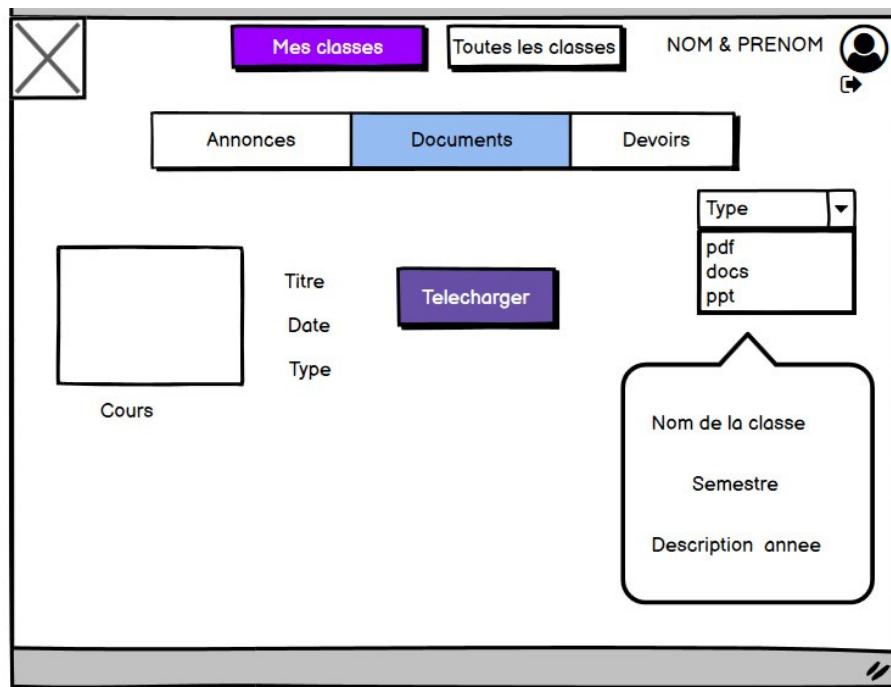


Figure 21 : Affichage des documents par type

**Cette figure affiche les documents de la classe choisie avec la possibilité de trier par date,par type ou par catégorie et de les télécharger**

## PARTIE DES DEVOIRS

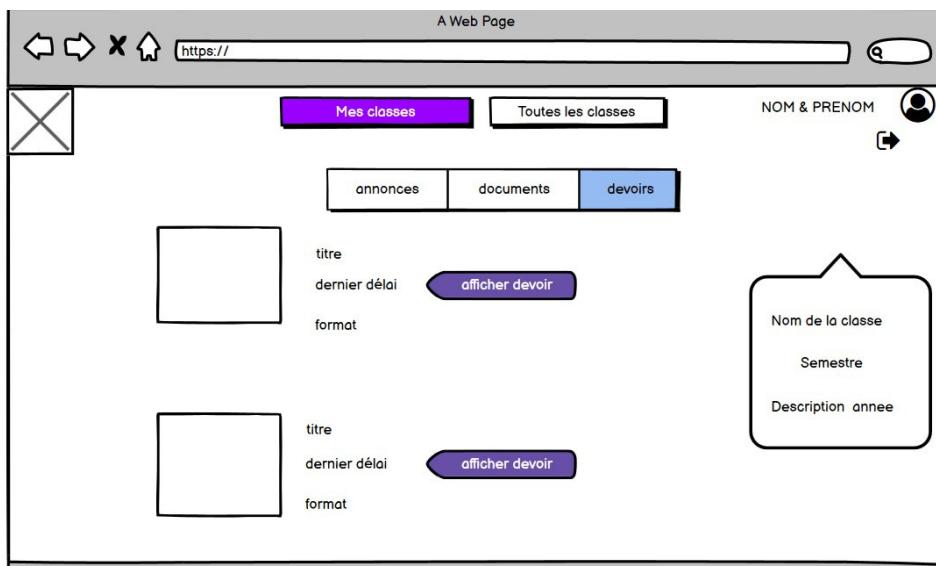


Figure 22: partie devoirs etudiant

**Cette figure affiche les devoirs à faire**



A Web Page

https://

Mes classes Toutes les classes NOM & PRENOM

énoncé dernier délai format demander

Ajouter

travail Modifier

Nom de la classe  
Nom de devoir  
Semestre  
Description année

Ajouter votre travail

importer

valider

Figure 23: partie ajout des devoirs

**Cette figure permet à l'étudiant de poster son travail avant la fin du délai .sinon , il n'aura pas la possibilité d'ajouter ou de modifier son travail**

# L'espace enseignant:

The screenshot shows a web browser window with the title 'A Web Page'. At the top, there are navigation icons (back, forward, stop, home) and a search bar with the URL 'https://'. Below the header, there is a logo icon labeled 'LOGO', a user profile icon with 'NOM & PRENOM' next to it, and two buttons: 'Mes classes' (highlighted in purple) and 'Annonce'. A small search icon is also present. The main content area contains a form titled 'Creer une classe' with four input fields: 'Nom du cours', 'Nom de la formation', 'Semestre', and 'Anneee universitaire'. At the bottom of the form are two buttons: 'vider' and 'Creer'.

Figure 24: formulaire d'ajout d'une classe

**Cette figure permet à l'enseignant de remplir un formulaire pour créer une classe .**

The screenshot shows a web browser window with the title 'A Web Page'. At the top, there are navigation icons (back, forward, stop, home) and a search bar with the URL 'https://'. Below the header, there is a logo icon labeled 'LOGO', a user profile icon with 'NOM & PRENOM' next to it, and five buttons: 'Mes classes' (highlighted in purple), 'Annonce' (highlighted in blue), 'Document', 'Devoir', 'Absence', 'Note', and 'Demande'. A small search icon is also present. The main content area displays two existing announcements with fields for 'Titre', 'Description', and 'Date', each accompanied by a delete icon. To the right, there is a form titled 'Ajouter une annonce' with fields for 'Titre', 'Date' (with a calendar icon), and 'Description', followed by a 'publier' button.

Figure 25: partie annonces enseignant

**Cette figure permet à l'enseignant de consulter, supprimer et de modifier une annonce et aussi de la créer .**

Figure 26: partie document enseignant

**Cette figure permet à l'enseignant de consulter, modifier et de supprimer un document et aussi de l'ajouter .**

Figure 27: partie devoir enseignant

**Cette figure permet à l'enseignant de consulter et de supprimer un devoir et aussi de le créer et le publier .**

Cne	Prenom	Nom	Date absence
A123456788	Aya	Moufid	<input checked="" type="checkbox"/> present
V12345678	chaimaa	souhir	<input type="checkbox"/> present
R123512763	Kawtar	famrouji	<input checked="" type="checkbox"/> present

Figure 28: partie absence

**Cette figure permet à l'enseignant de gérer l'assiduité des étudiants d'une classe par séance (Date précise).**



A Web Page

https://

		Mes classes		Annonce			
		Annonce	Document	Devoir	Absence	Note	Demande
<b>Creer controle</b>							
Cne	Prenom	Nom	Note_devo	Note_norm	Note_rattrapag	Note_control	Note_assiduit
A123456788	Aya	Moufid	20	20		20	20
V123245678	chaima	souhir					
R123512763	Kawtar	farrouj					
<b>Coefficients</b>							
coff_Note_devoir	0,25						
coff_Note_assiduite	0,5						
coff_Note_normal	0,75						
coff_Note_rattr	0						
coff_Note_cont	0						
<b>Note_reference</b> 15 <b>Generer note assiduite</b>							
<b>Calculer note</b>							

Figure 29: partie notes

**Cette page permet à l'enseignant de demander le calcul des notes globales d'une classe. Il aura les choix des notes à utiliser dans le calcul : Assiduité ,Devoirs , Contrôles ,Examen (sessions normale et rattrapage ) .**

A Web Page

https://

		Mes classes		Annonce			
		Annonce	Document	Devoir	Absence	Note	Demande
	CNE	Date				<b>Accepter</b> <b>Refuser</b>	
	Nom prenom					<b>Accepter</b> <b>Refuser</b>	
	Email					<b>Accepter</b> <b>Refuser</b>	
	CNE	Date				<b>Accepter</b> <b>Refuser</b>	
	Nom prenom					<b>Accepter</b> <b>Refuser</b>	
	Email					<b>Accepter</b> <b>Refuser</b>	

Figure 30: partie demandes

**Cette page permet à l'enseignant d'accepter ou refuser les demandes des étudiants pour accéder à une classe.**

# Conclusion



**Dans ce chapitre on a présenté notre application Web par les différents modèles et diagrammes pour donner une idée globale pour rendre notre projet plus claire et compréhensible sur les différentes cotés.**

**Toutes ces tâches vont être traduites sur des interfaces qu'on va représenter dans le chapitre suivant.**



# CHAPITRE III :

## Application

**Dans ce chapitre, nous présentons l'environnement matériel et logiciel du projet. Ensuite, nous nous intéressons à la description de quelques interfaces du système implémenté dans le cadre de quelques scénarios d'utilisation**

# APPLICATION

## I. Environnement de travail :

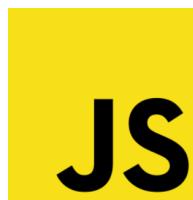
### 1. Environnement Hard :

- Hôte : hp -PC
- Microprocesseur : Intel(R) Core(TM) i5 CPU M 520 @  
-RAM : 4,00 GO

## 2. LANGUAGES ET METHODES DE PROGRAMATION



**PHP** est un langage de script côté serveur. Il est utilisé pour développer des sites web statiques ou dynamiques ou des applications web. PHP signifie Hypertext Pre-processor, qui signifiait auparavant Personal Home Pages. Il est courant que ce langage soit associé à une base de données, tel que MySQL.



**JavaScript** est un langage informatique utilisé sur les pages web. Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer. La particularité du JavaScript consiste à créer des petits scripts sur une page HTML afin de permettre des traitements (calculs, animations, ...) qui ne peuvent être effectués par le langage HTML. Pour y conclure, l'intérêt du JavaScript est d'exécuter un code sans avoir à recharger une nouvelle fois la page.



**MySQL** est un véritable serveur de base de données SQL Multiutilisateur et Multi-threaded ce qui signifie qu'il peut utiliser plusieurs CPU. IL permet de créer et de gérer des bases de données relationnelles. La nature de ce SGBD a permis son extension pour devenir la première base de données dans le monde - même si MySQL est encore en développement, il propose déjà un ensemble de fonctionnalités riches et extrêmement utiles sous Linux et Windows. Pour la réalisation e notre base nous avons utilisé MySQL grâce à sa fiabilité et sa disponibilité gratuitement sous une licence libre et il possède une interface intuitive et conviviale avec PHP et il est aussi très suffisant pour une base de données tel que celle qu'on va la réaliser

### **3. Design & Multimédia :**



**HTML5**, pour HyperText Markup Language 5, est une version du célèbre format HTML utilisé pour concevoir les sites Internet. Celui-ci se résume à un langage de balisage qui sert à l'écriture de l'hypertexte indispensable à la mise en forme d'une page Web.



Le terme **CSS** est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

### **4. Outils utilisés**



Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppression et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier SQL et d'y transférer ses données, même sans connaître SQL.

C'est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP.



**Sublim text** Un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

**Sublime Text** intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l'auto complétion, un système de plugins



**Balsamiq** est un logiciel de conception de wireframes qui permet aux équipes de créer des maquettes et des prototypes interactifs, mais aussi de réaliser des tests utilisateurs. L'outil est destiné à tous ceux qui ont besoin de créer des wireframes, et son utilisation ne requiert pas d'expérience particulière dans le webdesign.

Le logiciel est basé sur un système de glisser-déposer, qui permet aux utilisateurs de créer des interfaces et des applications pour le web et les mobiles de manière très simple.

Balsamiq offre un large choix d'éléments UX pour personnaliser au maximum les wireframes.



**SAP PowerDesigner** intègre différents niveaux de la modélisation des données, conceptuel, logique et physique, afin d'aider la représentation dimensionnelle des projets décisionnels.

Au sein de l'environnement uniifié de PowerDesigner, les chefs de projets, concepteurs et administrateurs de données pourront trouver différents modèles et techniques de modélisations spécifiques apportant des fonctionnalités adaptées à chaque processus métier.

SAP PowerDesigner propose également de créer une vision globale, dynamique et unique du système d'information de l'entreprise permettant d'associer des données sources aux projets cibles. Il permet également de garantir la traçabilité des flux d'informations dans les applications et de favoriser le partage et l'échange d'informations en toute sécurité.



**Apache est le serveur web le plus répondu permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers en format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur distant. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.**

# Introduction au modèle MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre

pour structurer une application. Dans l'architecture MVC, les rôles des trois entités sont les suivants.

**modèle : données (accès et mise à jour)**

**vue : interface utilisateur (entrées et sorties)**

**contrôleur : gestion des événements et synchronisation**

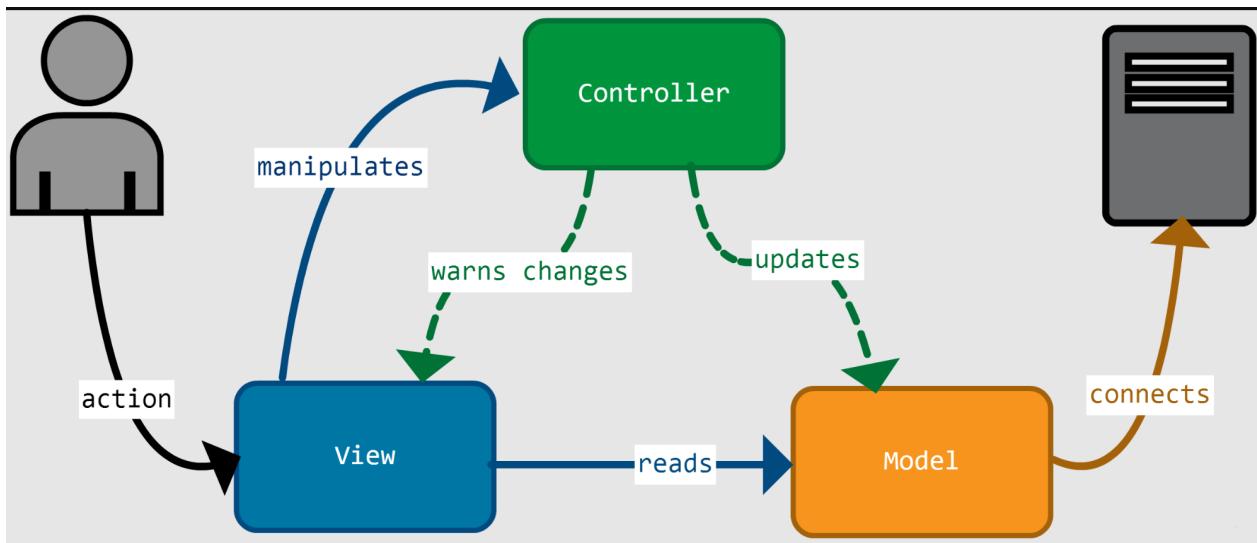


Figure 31: MVC

## 1-Modèle

Il décrit les données manipulées par l'application. Dans le cas typique d'une base de données, le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression et modification). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

On peut donc conclure que c'est le modèle qui contient toute la logique métier de l'application.



## 2-Vue

**La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir tous les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, ...). Ses différents événements sont envoyés au contrôleur.**

**La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données. Par exemple, l'application de conversion de bases a un entier comme unique donnée. Ce même entier est affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.**

## 3-Contrôleur

**Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour qu'elle se mette à jour. On conclut donc que le contrôleur est l'intermédiaire entre la vue et le modèle.**

## II. Principales interfaces graphiques

**Cette partie dénombre la présentation des scénarios applicatifs de l'application. Nous disposons de deux volets : enseignant et étudiant. Nous allons présenter dans ce qui suit, les imprimés-écran des principales interfaces réalisées dans notre application .**

## Interface d'accueil de l'application:

**C'est une introduction à notre application pour présenter les différents services offerts.**



**Figure32: interface d'accueil**

## Les interfaces côté enseignant:

### **Interface d'inscription:**

**L'interface ci-dessous présente le formulaire d'inscription client, dans le cas où le client oublie de saisir les champs obligatoires, un message d'erreur s'affiche pour la vérification.**

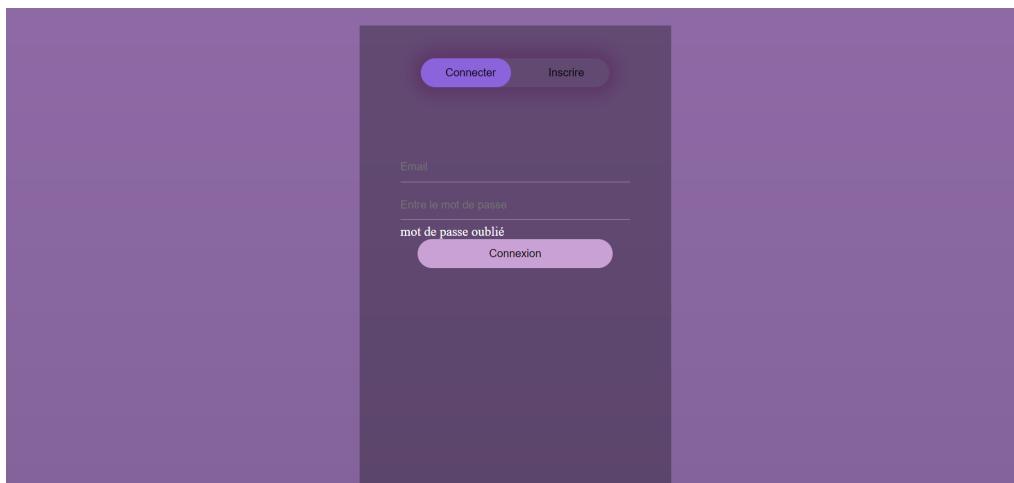
The screenshot shows a registration form for teachers. At the top right are two buttons: 'Connecter' and 'Inscrire'. Below them is a radio button group for selecting 'ETUDIANT' or 'ENSEIGNANT'. The 'ENSEIGNANT' option is selected. The form fields include: 'Nom' (Name), 'Prenom' (First Name), 'Email', 'Entre le mot de passe' (Enter password), 'le mot de passe' (password), 'mm/dd/yyyy' (date of birth), and a file input field for 'photo:' (photo). A note below the date field says 'photo: Choose File No file chosen'. At the bottom is a purple 'inscrire' (register) button.

**Figure33: interface d'inscription**



### Interface d'authentification :

**La figure ci-dessous représente la page d'authentification de l'enseignant. En effet,L'authentification est la première tâche qu'un administrateur doit effectuer pour accéder au menu administrateur approprié. Dans le cas où il saisit des informations erronées un message d'erreur lui sera affiché à fin de corriger les informations: son email ou mot de passe.**



**Figure 34-Authentification**

### Interface d'accueil espace enseignant :

**La page d'accueil espace enseignant contient toutes les classes qu'il dirige et qu'il a la possibilité de les gérer et aussi de créer des nouvelles classes .**



**Figure35: interface page principale de l'enseignant**

## Interface gestion des annonces :

Cette interface nous présente les annonces ainsi les différents opérations que l'enseignant peut faire telles que l'ajout, la modification, la suppression des annonces

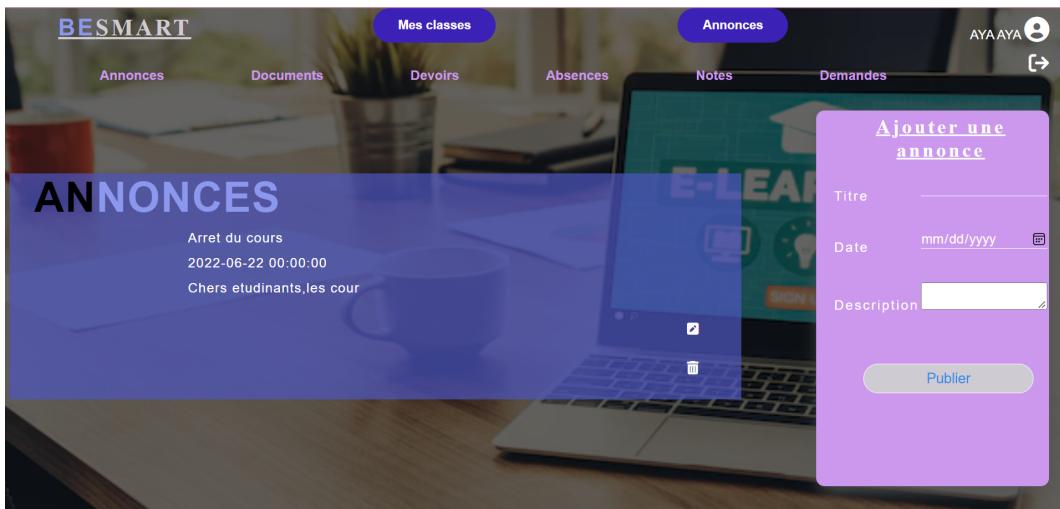


Figure36: interface gérer les annonces

## Interface gestion des documents :

Cette interface nous présente la partie document ainsi les différents opérations que l'enseignant peut faire telles que l'ajout, la modification, la suppression et la consultation des documents

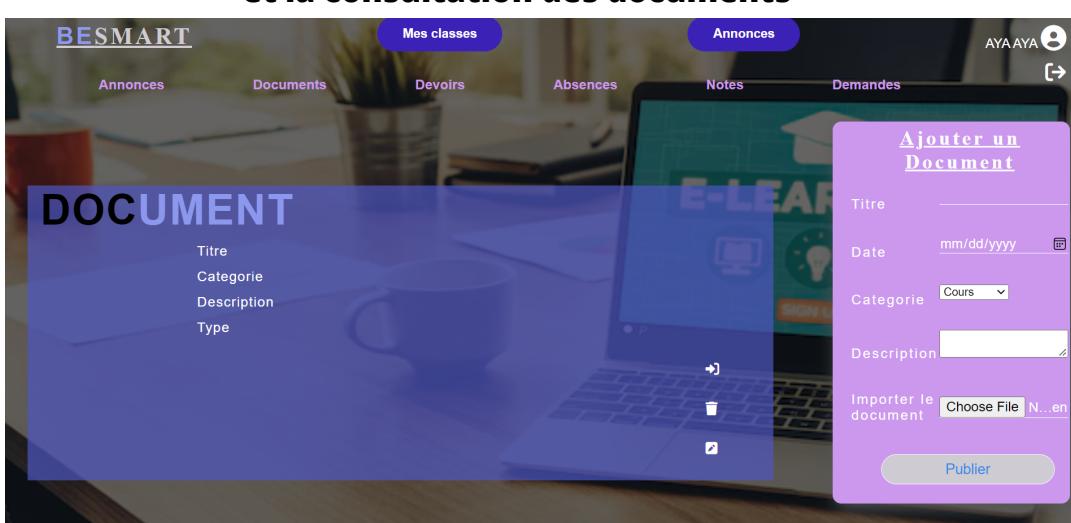


Figure37: interface gérer les documents

## Interface gestion des devoirs :

Cette interface nous présente la partie dans laquelle L'enseignant responsable peut programmer des devoirs pour une classe ainsi les différents opérations que l'enseignant peut faire telles que l'ajout, la modification, la suppression et la consultation de ces devoirs

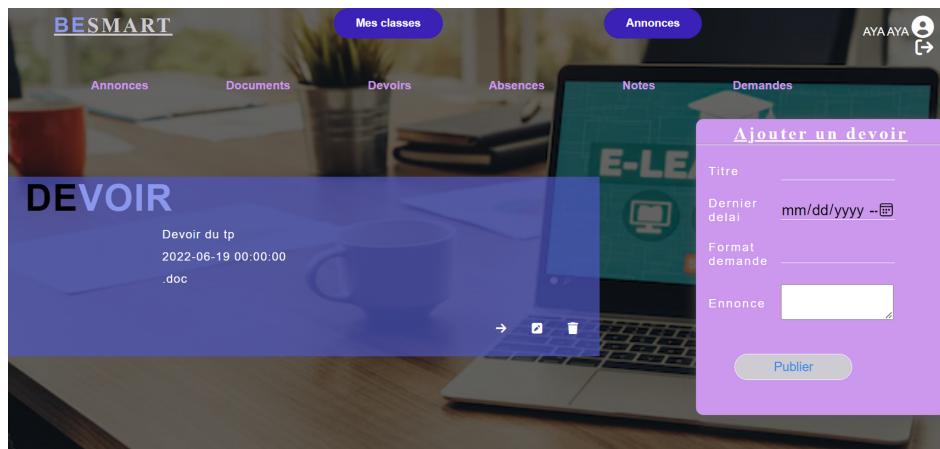


Figure38: interface gérer les devoirs

## Interface gestion des demandes :

A partir de l'interface ci-dessous, L'enseignant reçoit les demandes de la part des étudiants ,il peut accepter ou refuser ces demandes .

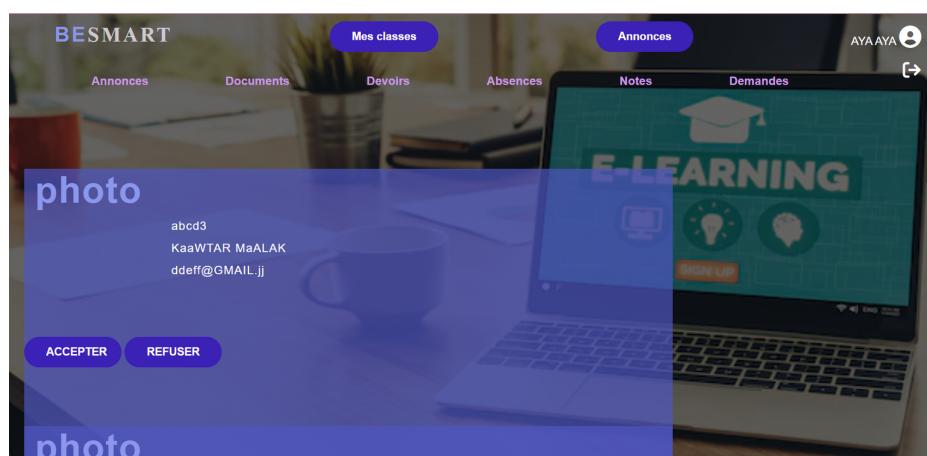


Figure39: interface gérer les demandes

## Interface gestion des absences :

A partir de l'interface ci-dessous ,L'enseignant responsable a la possibilité de gérer l'assiduité des étudiants d'une classe par séance (Date précise).

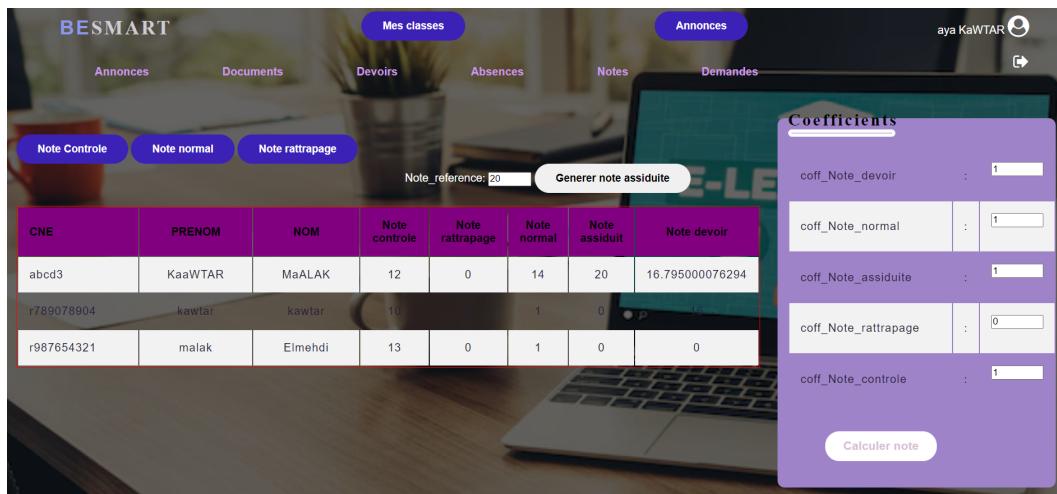


Figure40: interface gérer les absences

## Interface gestion des notes :

L'interface ci-dessous permet au enseignant de demander le calcul des notes globales d'une classe. Il aura les choix des notes à utiliser dans le calcul .

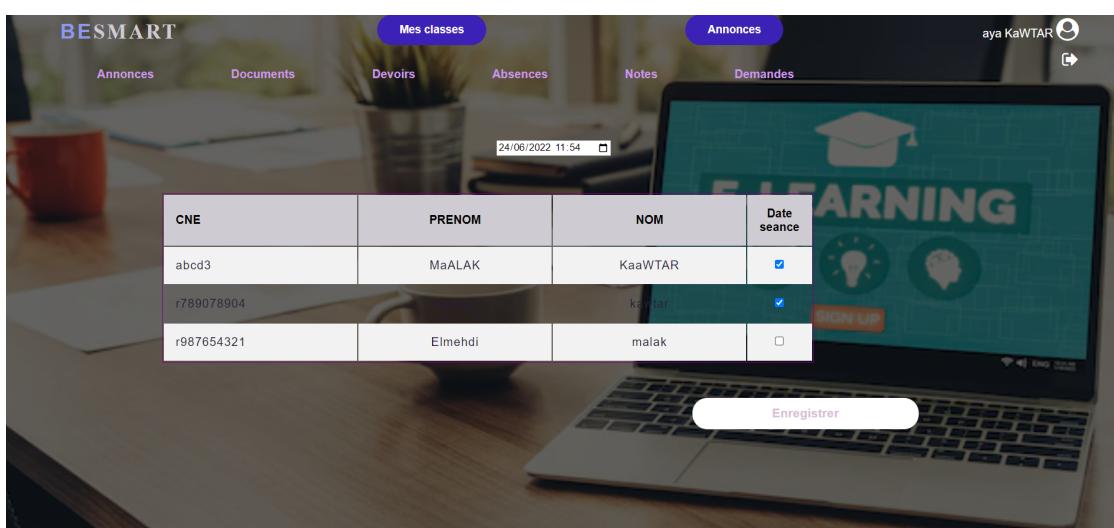


Figure41: interface gérer les nots

## Les interfaces coté étudiant:

### Interface d'inscription:

The screenshot shows a registration form for a student. It includes fields for CNE, Nom, Prenom, Email, Entre le mot de passe, le mot de passe, mm / dd / yyyy, and photo. There are also 'Inscrire' and 'Inscrire' buttons.

Figure42: interface d'inscription étudiant

### Interface d'accueil espace étudiant :

**les deux figures ci-dessous présente La page d'accueil de l'étudiant où il peut consulter toutes les classes dans lesquelles il est déjà inscrit ou cliquer sur le boutton toutes les classes dans laquelle il peut demander l'accès à plusieurs classes**

The screenshots show the student's main dashboard. The top one displays courses like C++ and Optique with a search bar and a 'Demande acces' button. The bottom one shows JAVA, HTML5, and DBA with similar course details and access request buttons.

Figure43: interface page principale de l'étudiant

## Interface Annonces :

Cette page est dédiée aux annonces qui a posté le prof

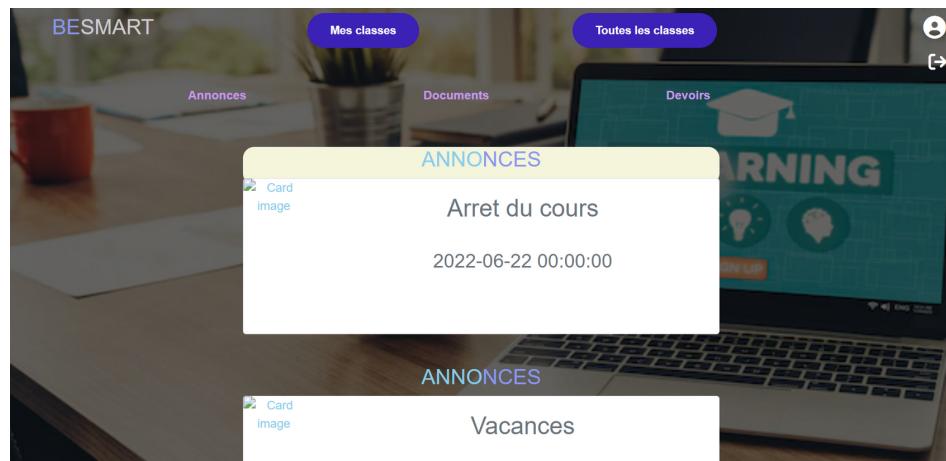


Figure44: interface annonces

## Interface Documents :

Cette page est dédiée aux document qui a posté le prof, où l'étudiant peut les telecharger et trier l'affichage

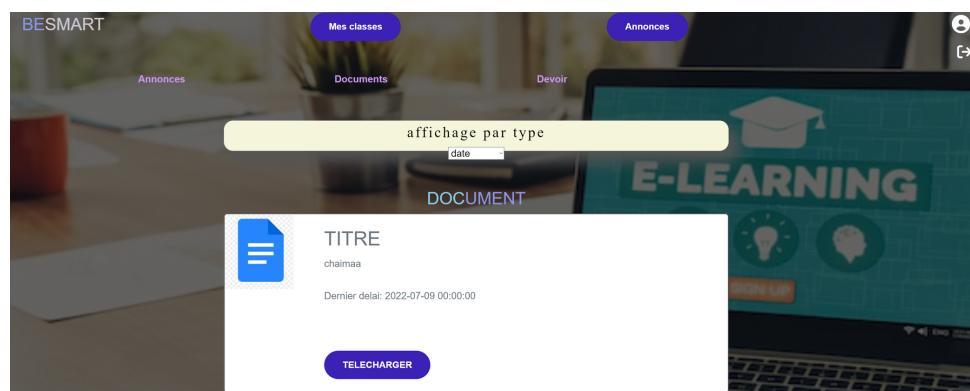


Figure45: interface document

## Interface Devoirs

Cette figure, représente la vue des devoirs pour chaque étudiants qui sera invité à les faire dans un deadline précis.

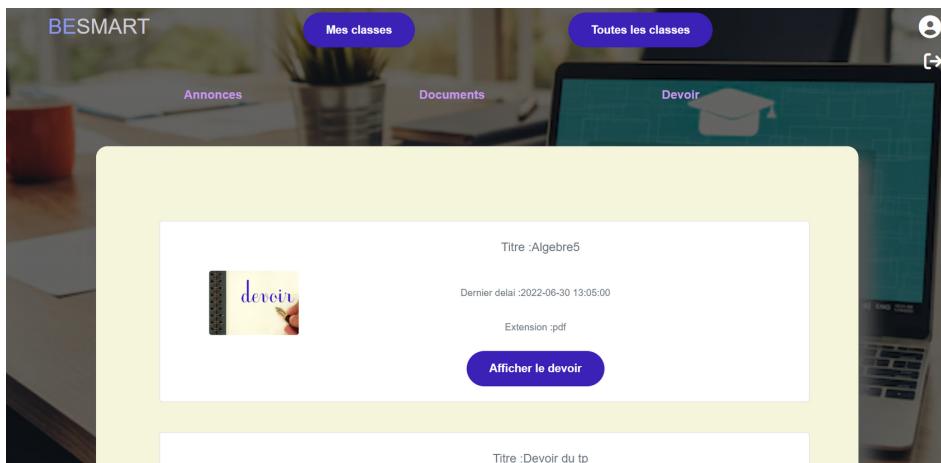


Figure46: interface devoirs

# Conclusion



**Cette partie s'est voulue une vision globale de la solution finale en abordant les technologies de développement utilisées, ainsi un ensemble des captures d'écrans pour une meilleure illustration.**

# Conclusion générale

Pour conclure ce rapport, nous pouvons affirmer que ce projet de fin d'études se situe dans le prolongement des années de formations suivies. Il plonge l'étudiant dans un monde novateur où la pratique dépasse souvent la théorie. Une capacité d'autonomie est mise en avant pour progresser et faire face à de nombreux problèmes. Ce projet a pour objectif de présenter les différentes étapes de conception de cette application web ainsi que ses différentes composantes .Cependant, ce projet constitue à la fois une étude théorique à travers la phase de conception et une étude pratique à travers la phase de réalisation de cette application .La réalisation de ce projet, nous a permis de créer un système de gestion des classes ,notre application BE SMART, ainsi que la conception, qui constitue une phase importante dans la représentation des différentes fonctionnalités de l'application .Cette application est extensible à être plus riche et on peut l'améliorer par plusieurs fonctionnalités qui facilite le contact entre les différents utilisateurs de l'application par exemple il est possible d'intégrer une boite de réception ou une créer une boite de messagerie instantanée .En effet, ce travail nous a permis d'un part d'enrichir nos connaissances théoriques à travers l'utilisation d'UML pour la conception orienté objet de systèmes d'informations, SQL comme un SGBD, et PHP comme un langage de programmation, d'une autre coté d'améliorer nos capacités de développement, afin d'en tirer profit dans le futur .A titre général, ce projet a constitué une expérience exceptionnelle pour nous . La vie universitaire ne permet que d'avoir un aperçu de son futur métier. On découvre rapidement que l'aspect pratique dépasse souvent les connaissances acquises et qu'il constitue une formation rapide.

# Bibliographie

<https://www.academia.edu/>

<https://www.docdoku.com/>

<https://www.canva.com/>

<https://www.php.net>

<https://www.w3schools.com/>

<https://datatables.net/>