

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 "Date de soutenance"

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



UML	UNIFIED MODELING LANGUAGE
MCD	CONCEPTUAL DATA MODEL
MPD	MODÈLE PHYSIQUE DE DONNÉES
MVC	MODEL VIEW CONTROL
CSS	CASCADING STYLE SHEETS
HTML	HYPertext Markup Language
JS	JAVASCRIPT
PHP	HYPertext Preprocessor
BD	BASE DE DONNÉES
SQL	STRUCTURED QUERY LANGUAGE

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.....	7
Listes des tableaux.....	8
Introduction générale.....	9

Chapitre I :Contexte Générale

Introduction.....	9
Objectifs.....	9
Missions du projet.....	9
Problématique.....	9
Livrables.....	9
Risques.....	9
Méthodologies de travail.....	9
Cycle de vie.....	9
Planning prévisionnel du projet.....	9
Planning réel du projet.....	9
Conclusion.....	9

Chapitre I I:Analyse et conception

Introduction.....	9
Besoins fonctionnels.....	9
Besoins non fonctionnels.....	9
Conception.....	9

Diagrammes des Cas d'utilisation

1-Definition.....	9
2-Description textuelle.....	9

Diagrammes des séquences

1-Definition.....	9
2-Diagramme de séquence : Authentification.....	9
3-Diagramme de séquence : Traitement de demande d'accès.....	,



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

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 4 chapitres : Le premier chapitre consistera à la présentation du projet, le contexte et les contraintes rencontrées.

Puis, le deuxième chapitre sera consacré à la définition de l'essentiel du travail en déterminant le plan suivit au développement de l'application et les outils utilisés.

Ensuite nous entamerons dans le troisième chapitre la partie modélisation en citant les différents utilisateurs, leurs besoins fonctionnels et les scénarios possibles des interactions avec l'application.

Après dans le quatrième et le dernier chapitre, nous allons 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 atténées.

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
Cahier des charges incomplet	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.
Le temps est court	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
Les pannes inattendues du Matériel	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.

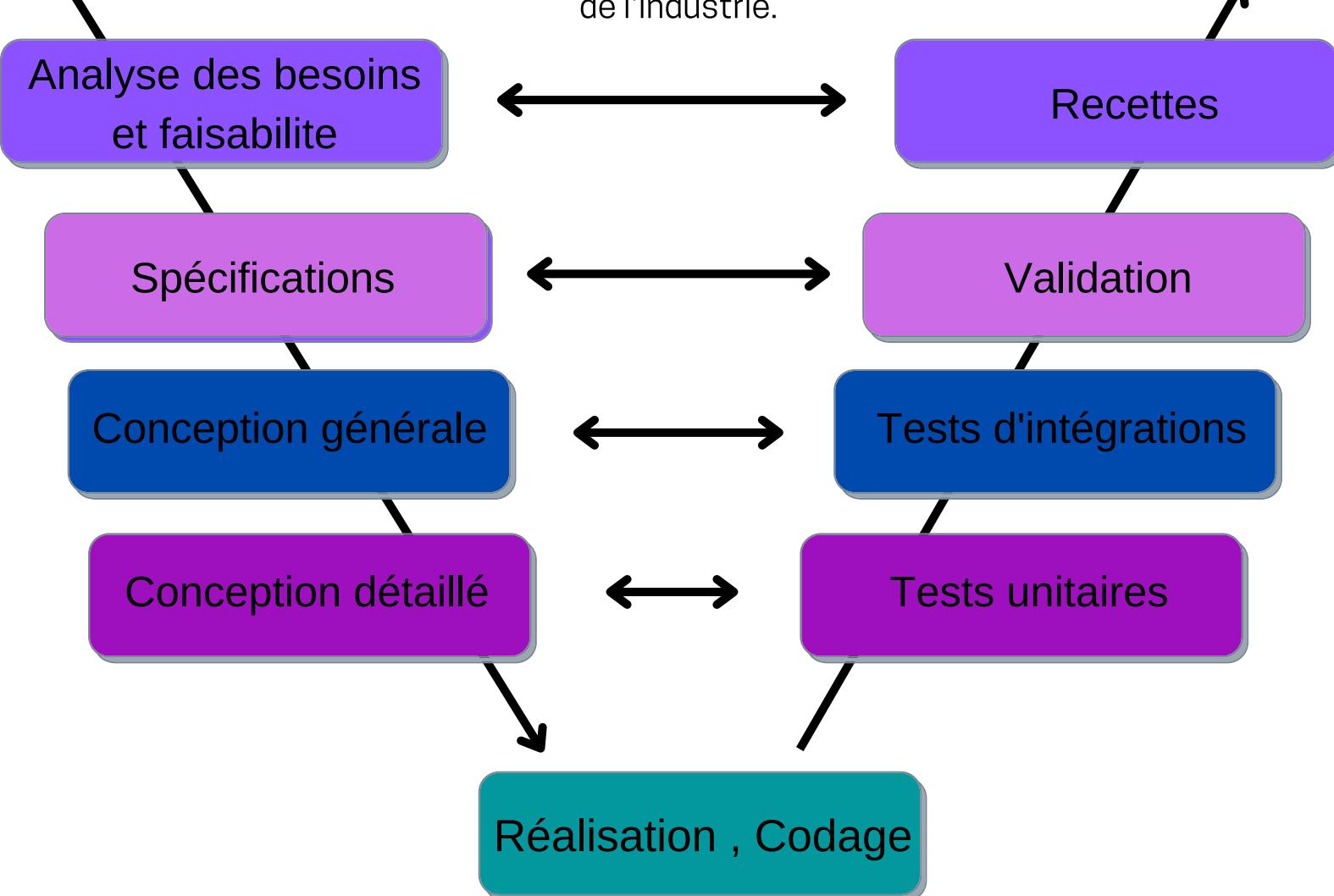


Figure 1:Cycle en V

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.

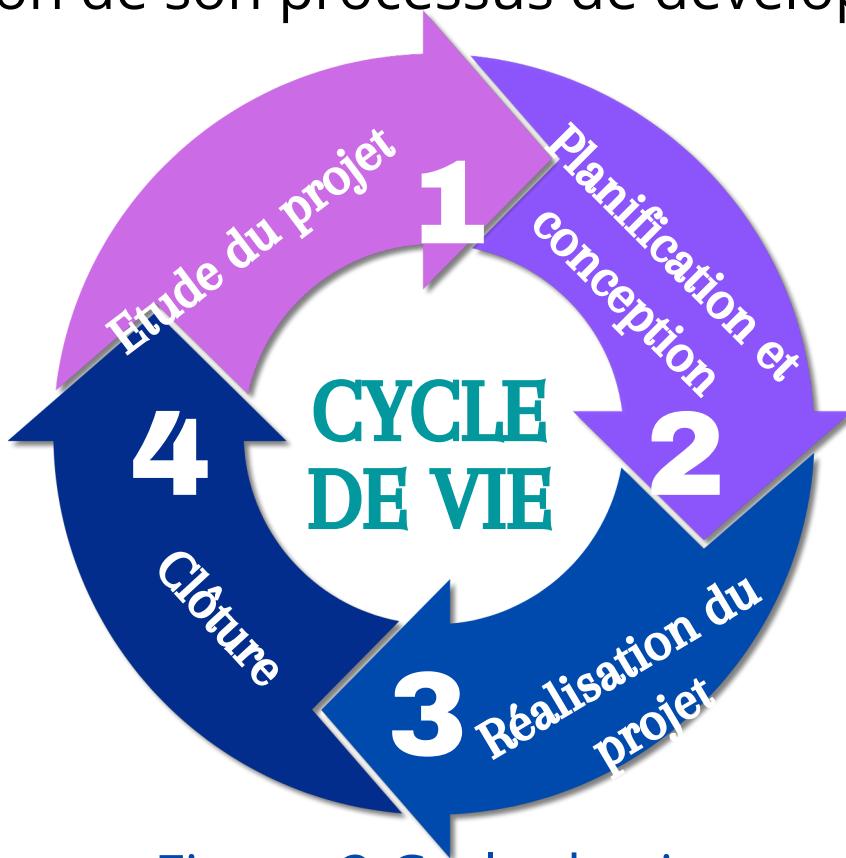


Figure 2:Cycle de vie



• 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.

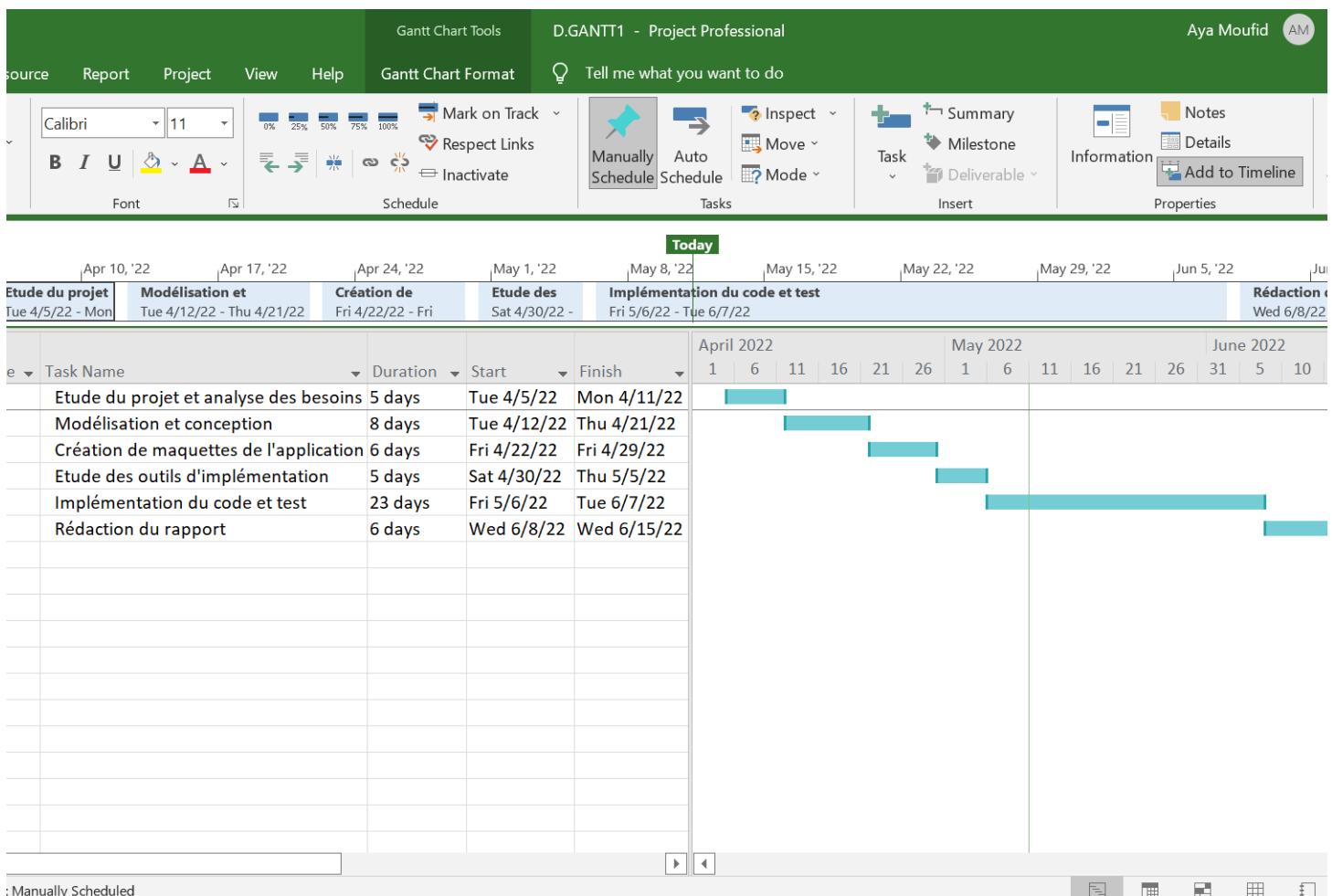


Figure3 : Diagramme de GANTT prévisionnel

Planning réel du projet

Finalement, 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.

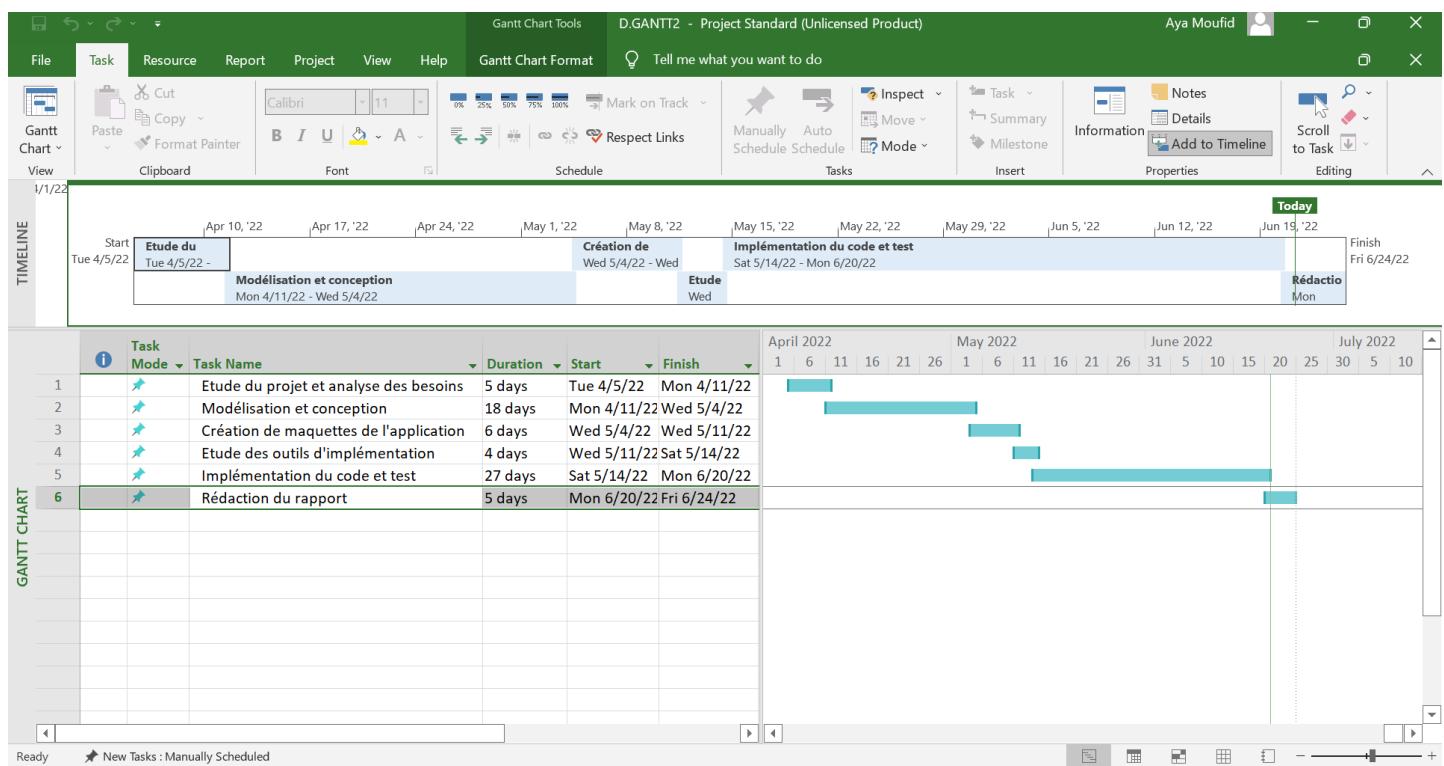


Figure4 :Diagramme de GANTT réel

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 .



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 téléchargé par l'étudiant en précisant les informations d'un document(Titre, Description, Date de mise en ligne, Categorie), et c'est à l'enseignant de choisir la catégorie du document.



4- Gestion des devoirs

Après la création de la classe, l'enseignant a 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 acteur qui sont l'enseignant et l'étudiant.

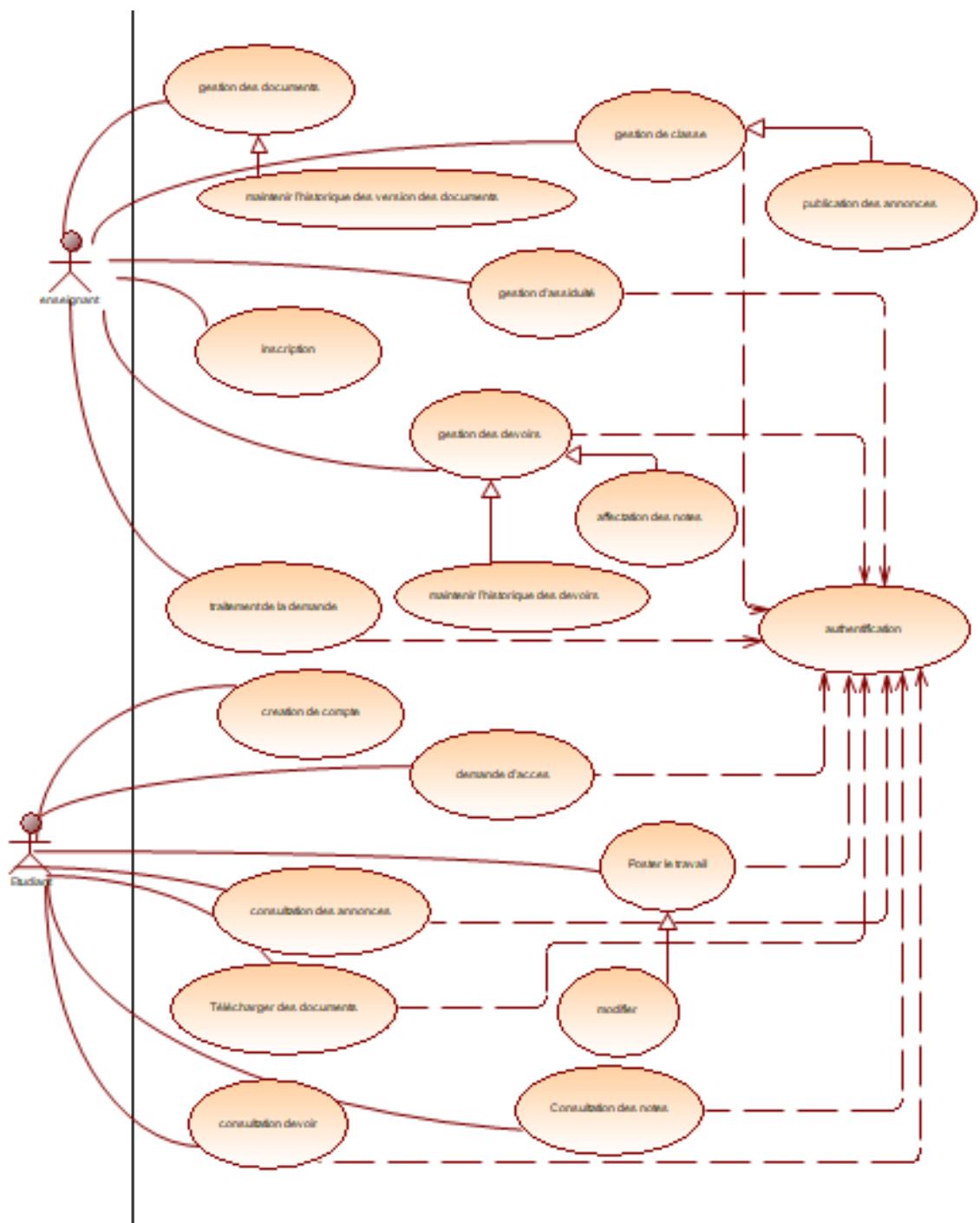


Figure 5: Diagramme de cas d'utilisation



2- Description textuelle

Acteurs concerné	Etudiant , Enseignant
Pré -condition	Compte existant
Post-condition	Autorisation d'accès accordée (L'utilisateur doit être créé dans la base de données et connaître ses identifiant)
Scénario nominal	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.
Scénario d'exception	1-L'utilisateur n'a pas saisi 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 »

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

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



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

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

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

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



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

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

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

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



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

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

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

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



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

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

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

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



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

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

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

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 sequence : Authentification

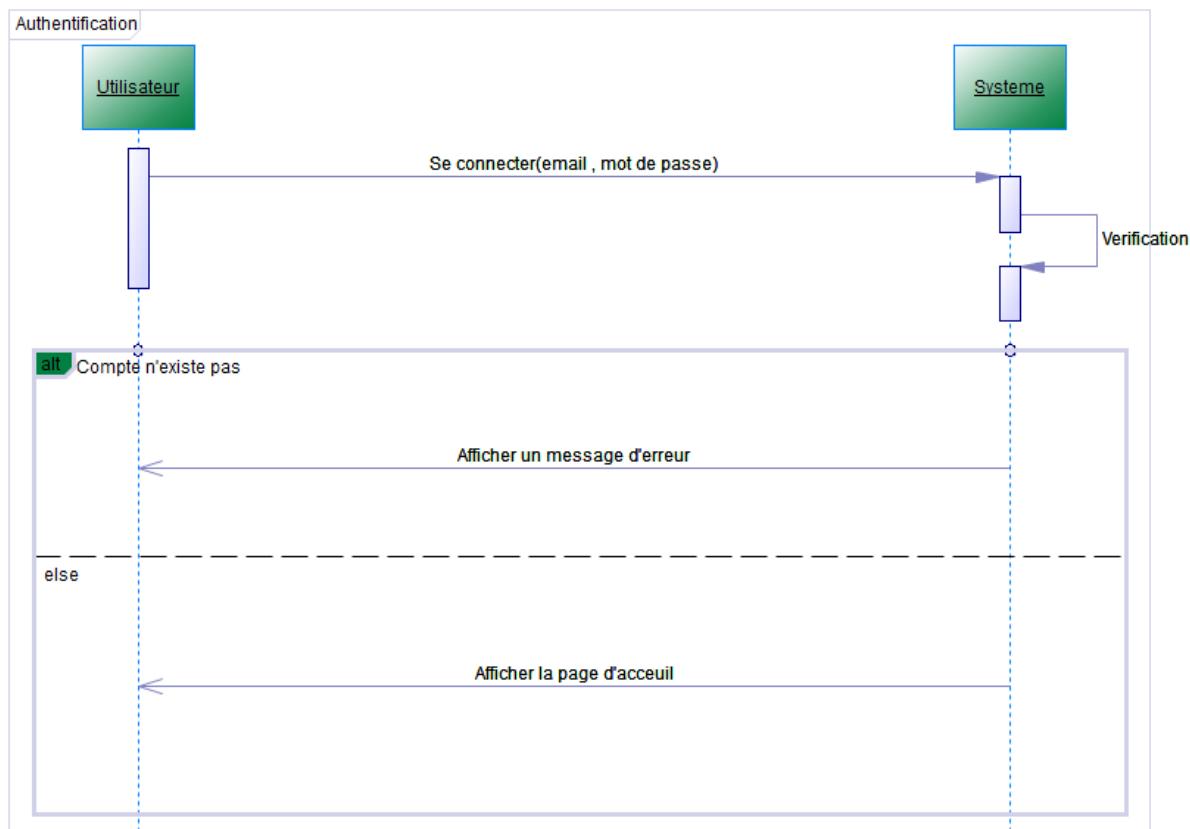


Figure 6: Diagramme de séquence Authentification

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 sequence : Traitement de demande d'accès

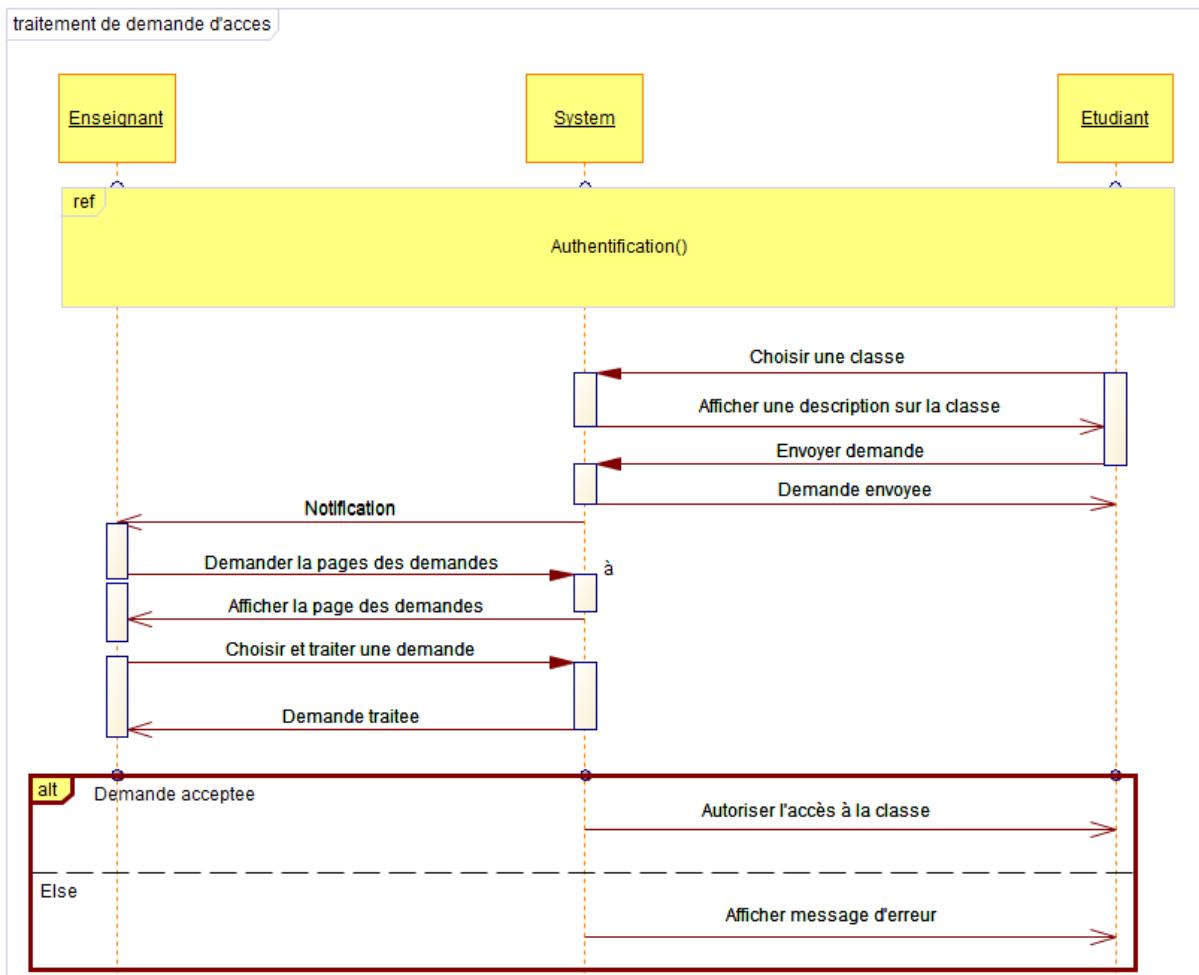


Figure7-Diagramme de sequence :Traitement de demande d'accès

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 sequence : Gestion des devoirs

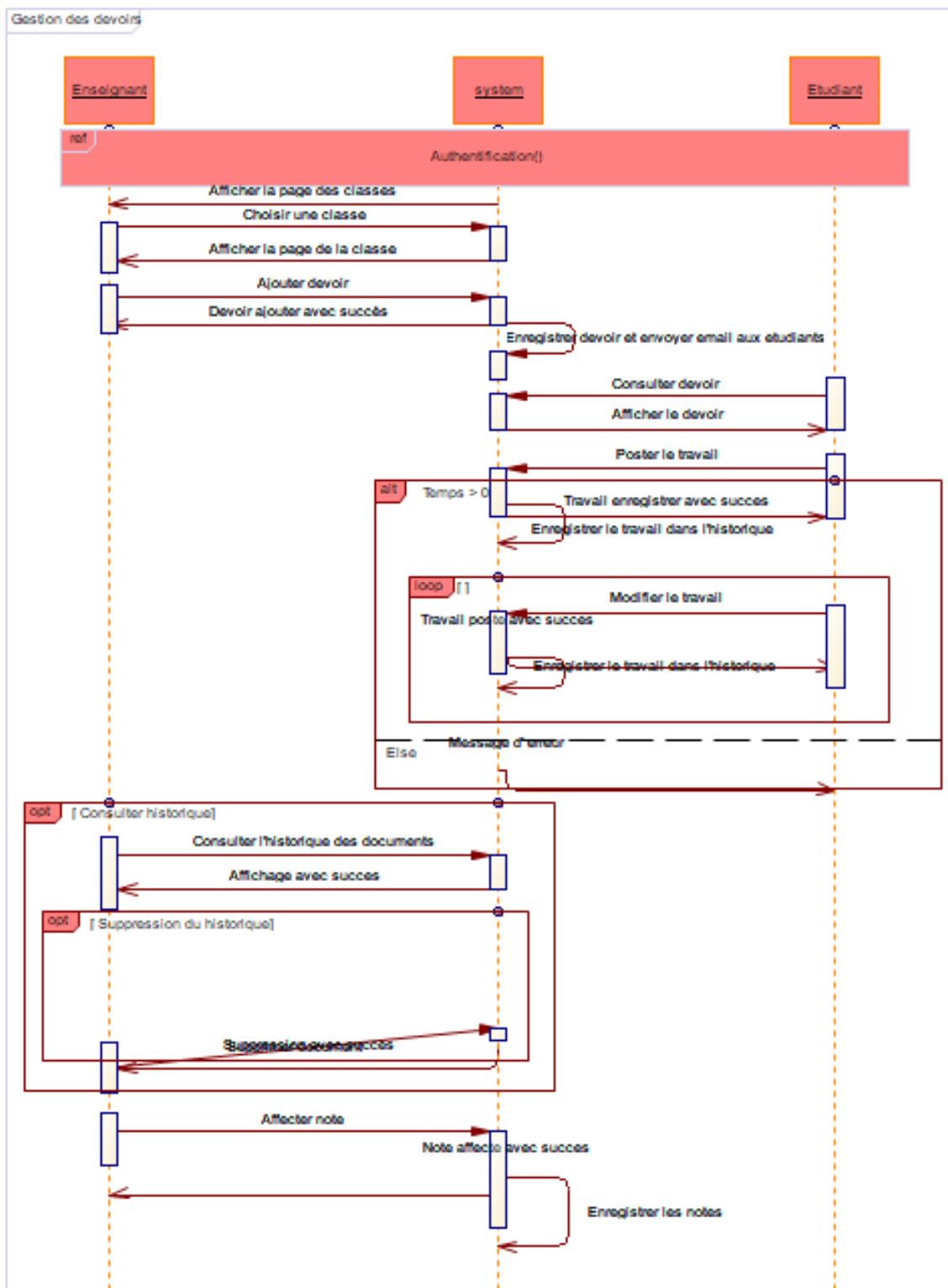


Figure 8 : Diagramme de sequence Gestion des devoirs



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

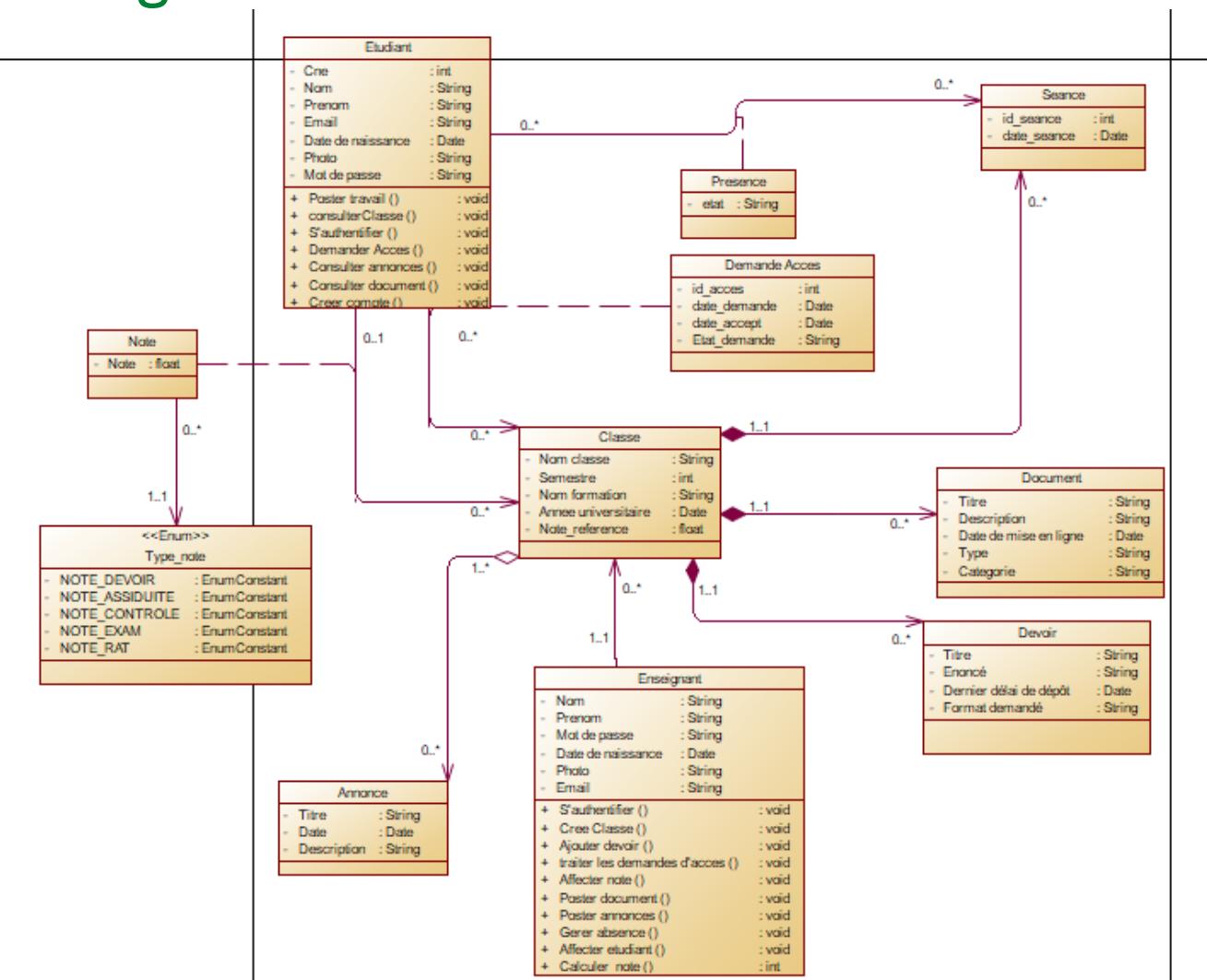


Figure9 : Diagramme de classe



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 etudiant qui a comme attributs nom, prenom, 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 qui est déjà inscrit dans, Poster travail () pour poster les devoirs programmés par l'enseignant et Consulter document () pour consulter les documents qui sont 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 etudiant 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, prenom, 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



Gerer absence() pour gérer l'assiduité des étudiants d'une classe par séance et Affecter etudiant() pour affecter un etudiant a une classe precise.

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 ,enoncé , dernier délai de dépôt (Date & Heure) , format demandé (Extensions possibles).

La classe annonce qui a comme attributs nom et date.

Et La classe seance qui a comme attributs id_seance, etat_assiduite, note_assiduite et note_reference.

On a utilisé le concept de **composition** car un devoir ,une seance 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 etudiant et classe dont les cardinalités 0..* et 0..* pour expliquer qu'un etudiant 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.

MCD

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

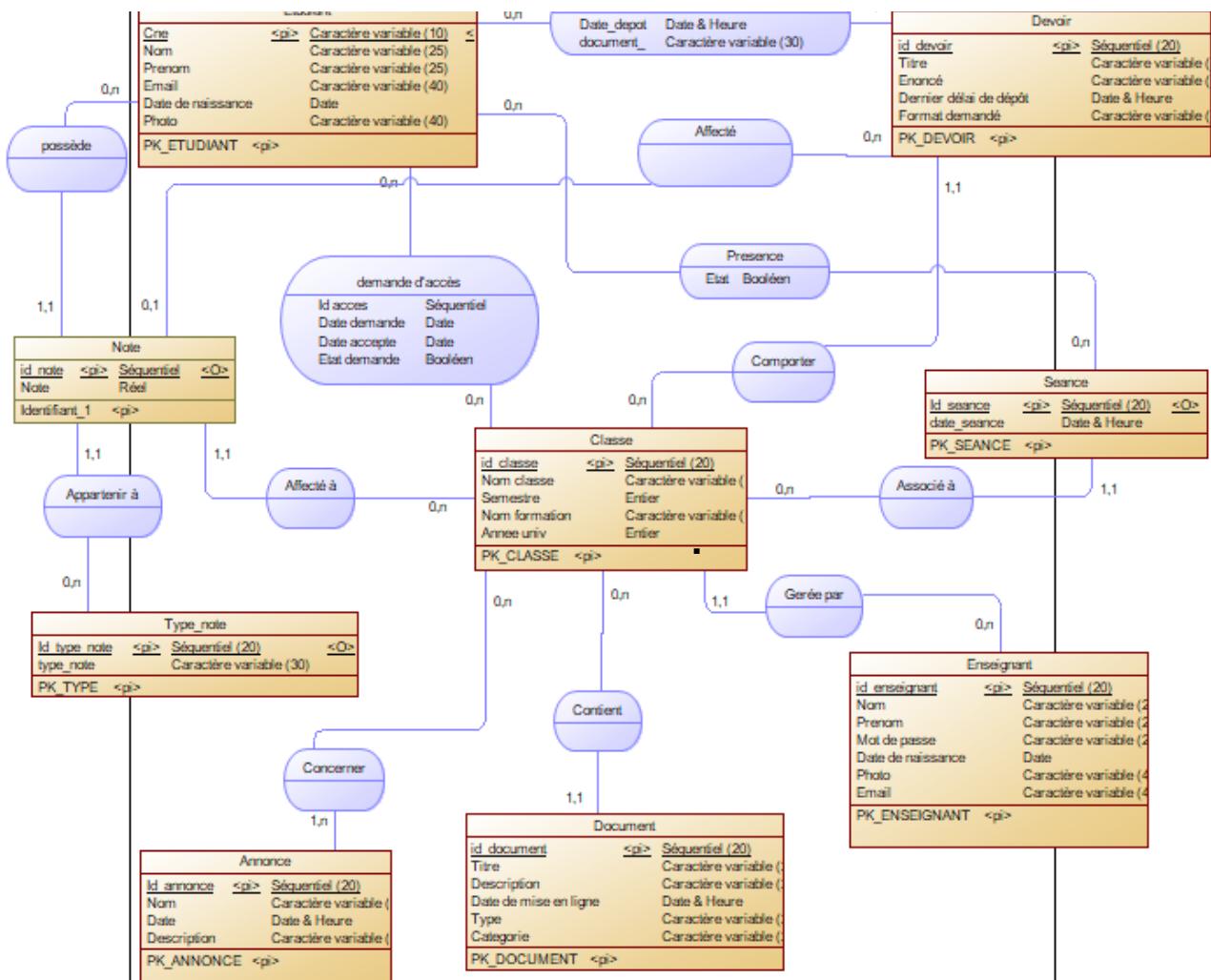


Figure 10 : Modèle conceptuel de données



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 etudiant peut ne pas déposer aucun devoir comme il peut éventuellement déposer plusieurs, sachant qu'il y a une association reliant les deux entités qui L'association depot possède comme attribut la date de depot du devoir de l'étudiant et le document poster par lui puisqu'elle dépond des deux à la fois. Notons à la fin qu un etudiant peut assiter à plusieurs seance.

Ensuite il peut demander acces a des divers classe comme il peut ne rien demander De même, l'association reliant entre eux contient tous les attributs dépondant des deux à la fois En ce qui concerne note elles ne peut être contenue que dans une seule classe et un seul etudiant.

Pour la classe il est identifiée par 'id classe', il peut etre gérer par un seul enseignant comme il peut contenir plusieurs etudiants,de plus chaque classe contient plusieurs devoirs,documents et des seances qui sont spécials pour elle ,n'oubliant pas qu' il contient des annonces chaque annonces peut concerner plusieurs classes.

Structure de la base de données

La figure 11 représente les tables utilisées dans notre base des données. Ces tables représentent l'unité de persistance de notre application

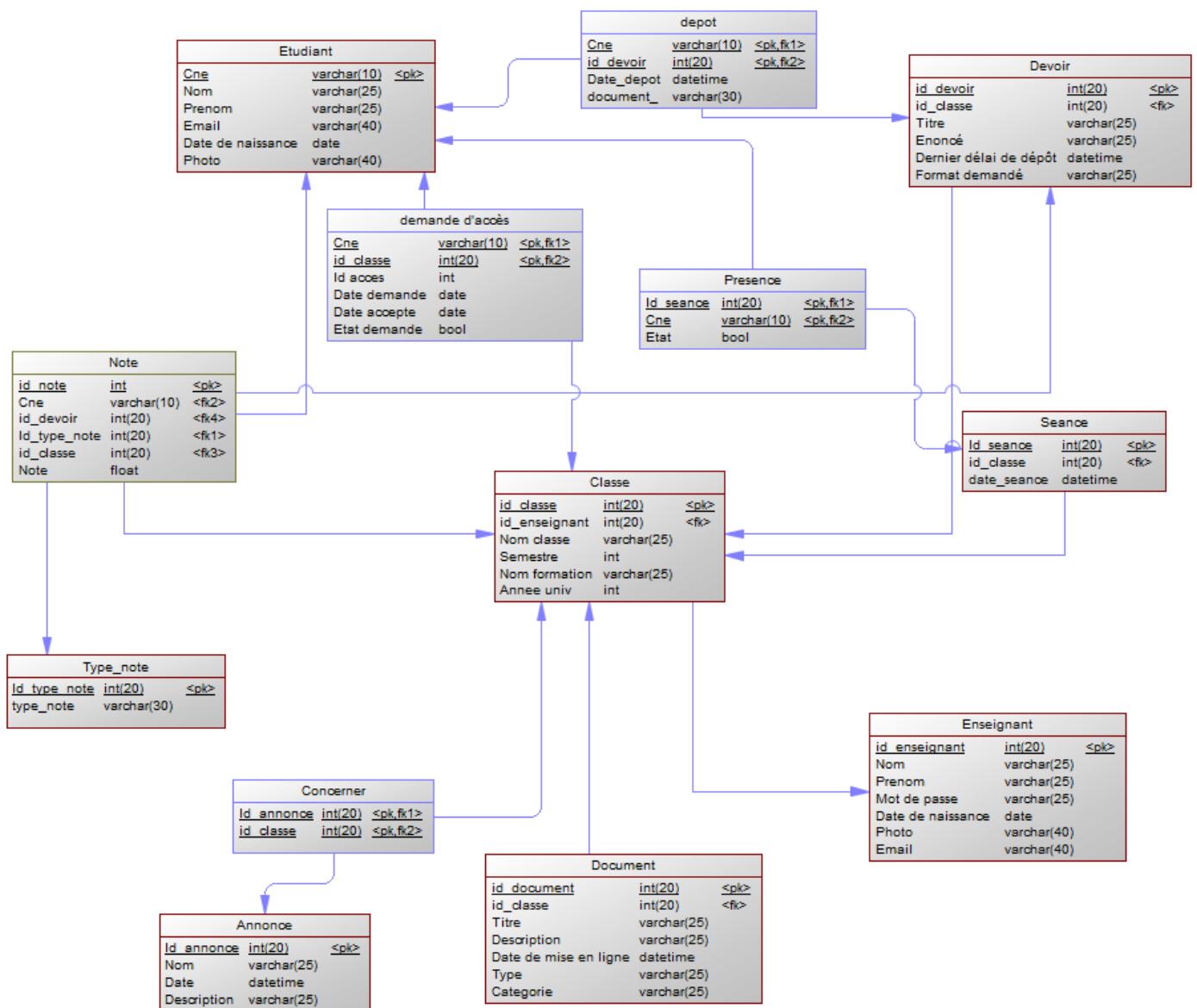
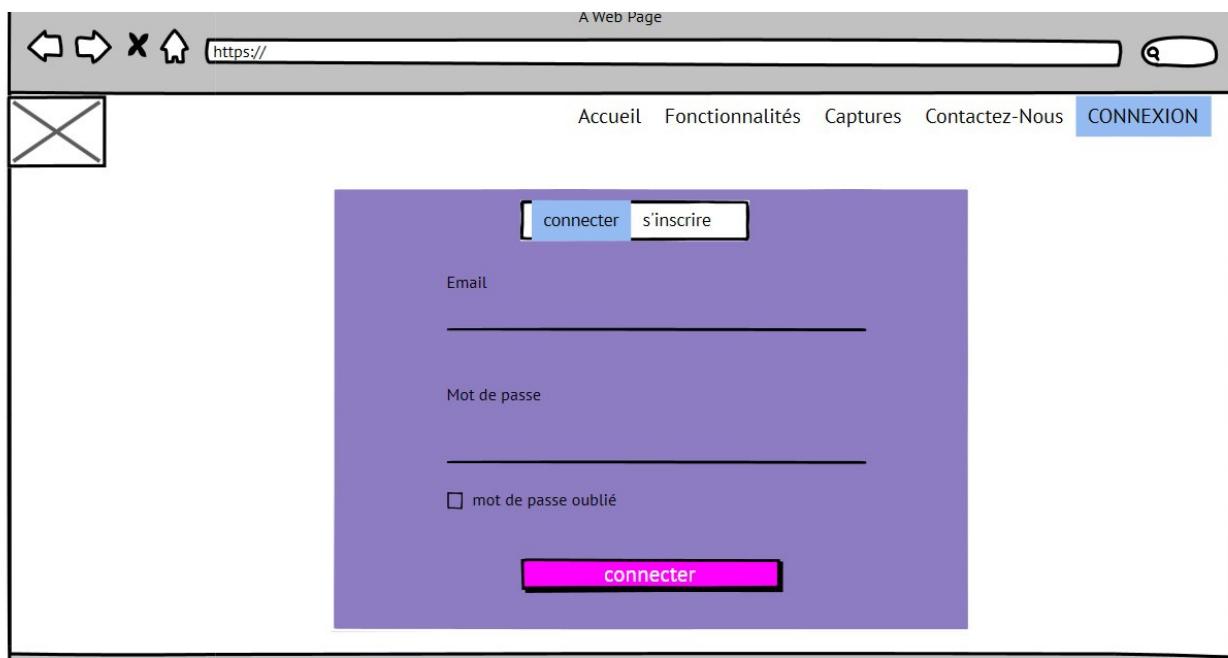


Figure 11: Modèle physique de données

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

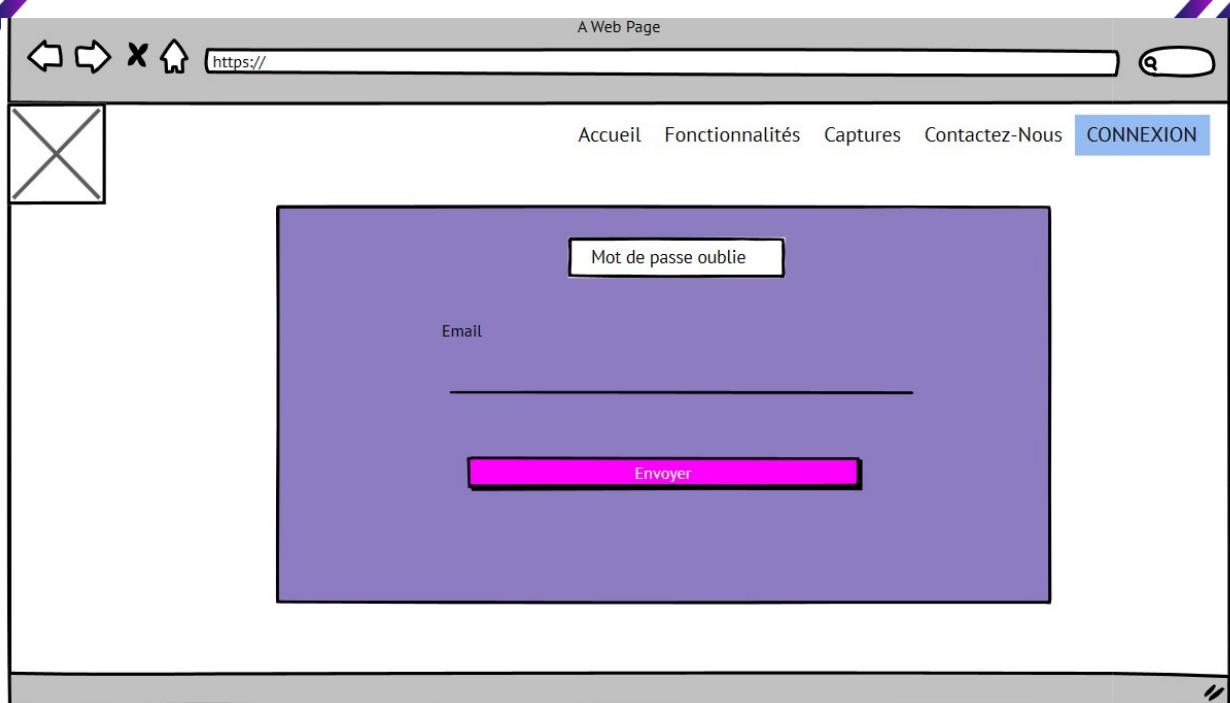


Figure 13:mot de passe oublié

Au cas où l'utilisateur oublie son mot de passe , on lui demandera d'envoyer 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

A screenshot of a web browser window showing an inscription form. The top navigation bar includes links for Accueil, Fonctionnalités, Captures, Contactez-Nous, and a blue "CONNEXION" button. On the left, there's a small window icon. The main content area has a purple background and contains a white box with two buttons at the top: "connecter" and "s'inscrire" (the latter is highlighted in blue). Below are several input fields: "Nom" (with "Etudiant" and "Enseignant" radio buttons, "Enseignant" is selected), "Prenom", "Email", "Mot de passe", "Confirmer mot de passe", "Date de naissance" (with a date input field "jj /mm / aaaa" and a calendar icon), "Photo" (with a "choisir un fichier" button and a message "Aucun fichier n'a été sélectionné"), and a pink "Inscrire" button at the bottom.

Figure 14:Inscription d'un nouveau enseignant

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 memes pages:

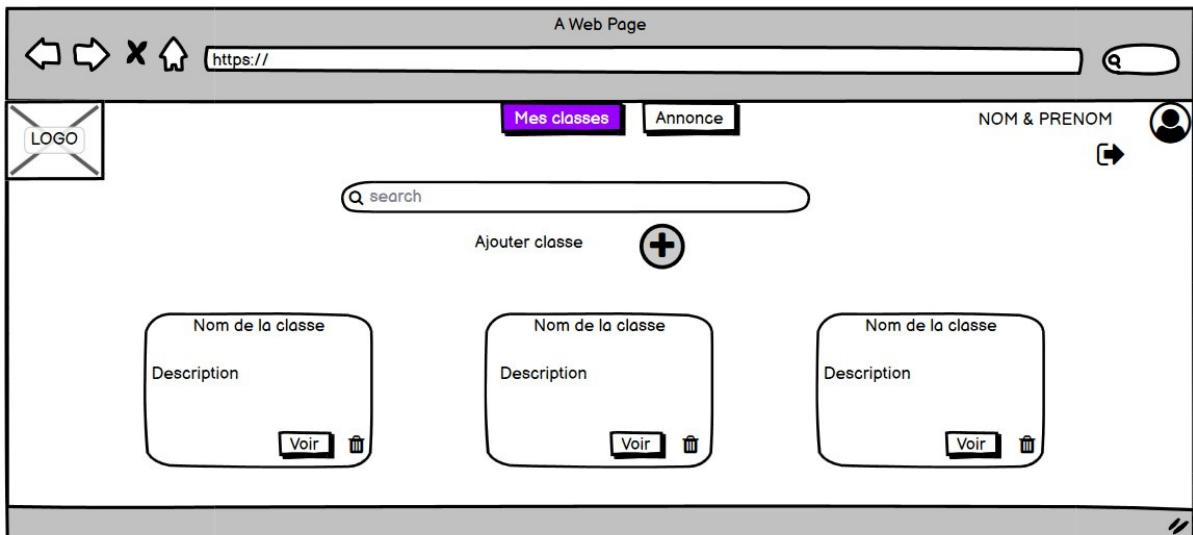


Figure 16:la page principale de l'enseignant

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

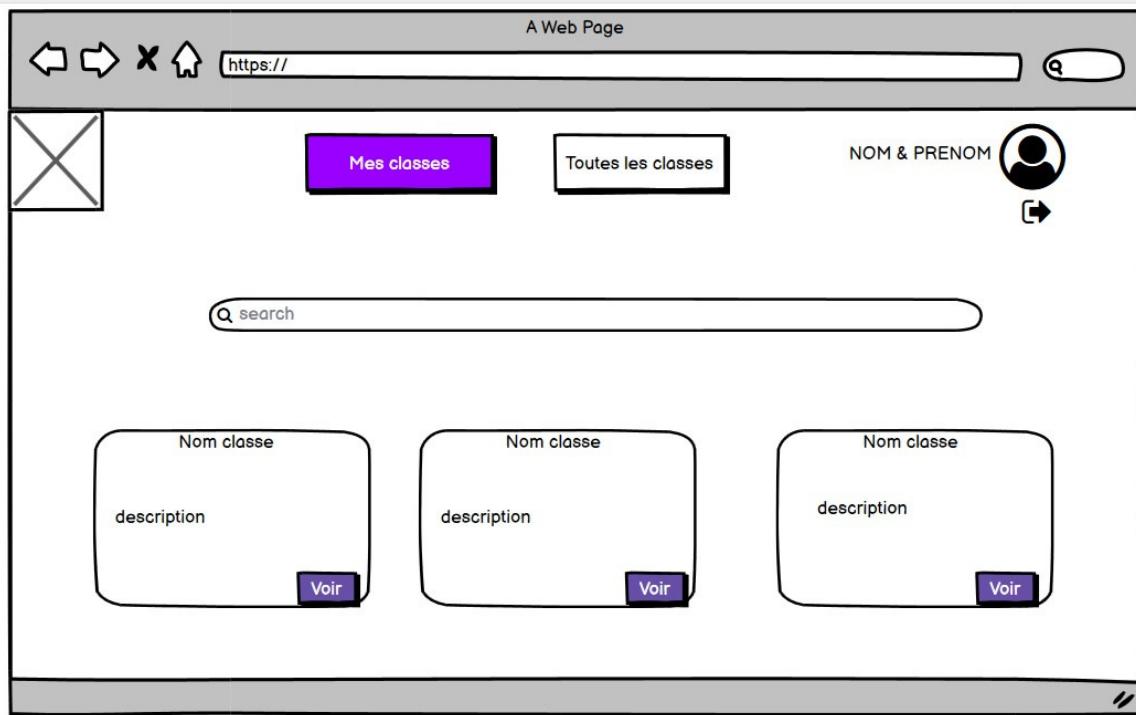


Figure 17: la page principale de l'étudiant

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

L'espace étudiant :

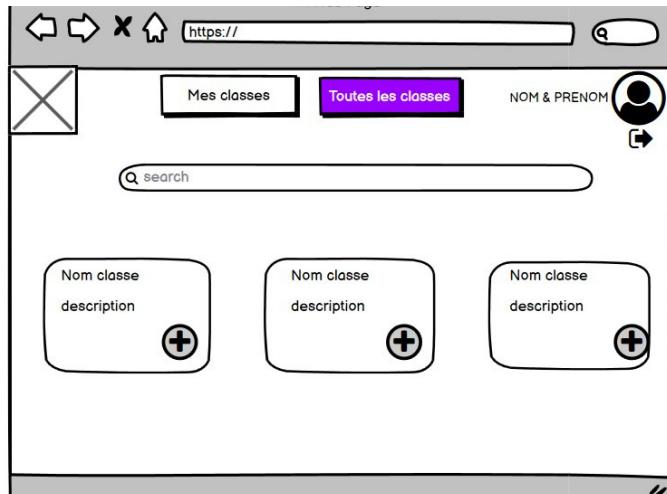


Figure 18: la pages des classes de chaque étudiant

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

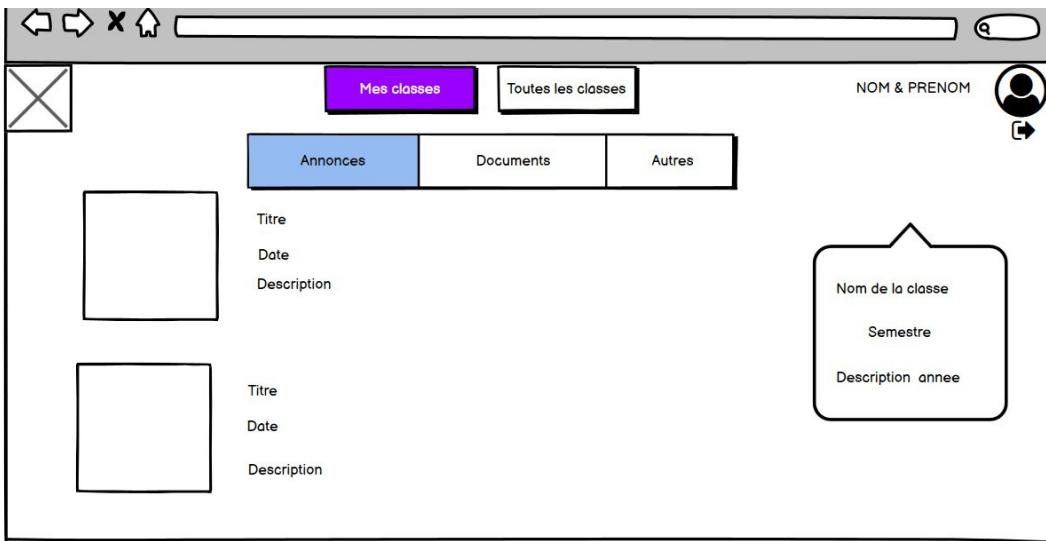


Figure 19: partie annonces

Cette figure affiche les annonces de la classe choisie avec les informations correspondants

PARTIE DES DOCUMENTS

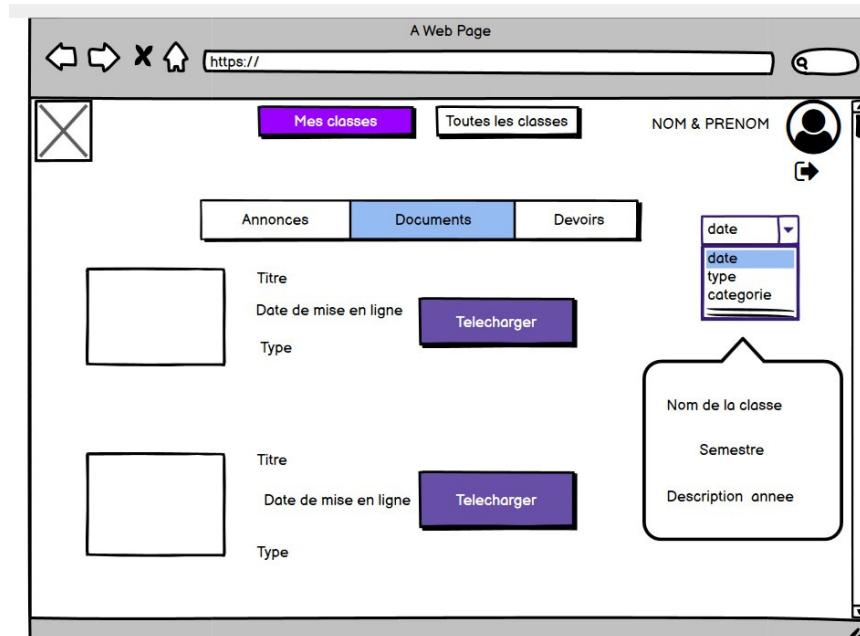


Figure 20: partie documents

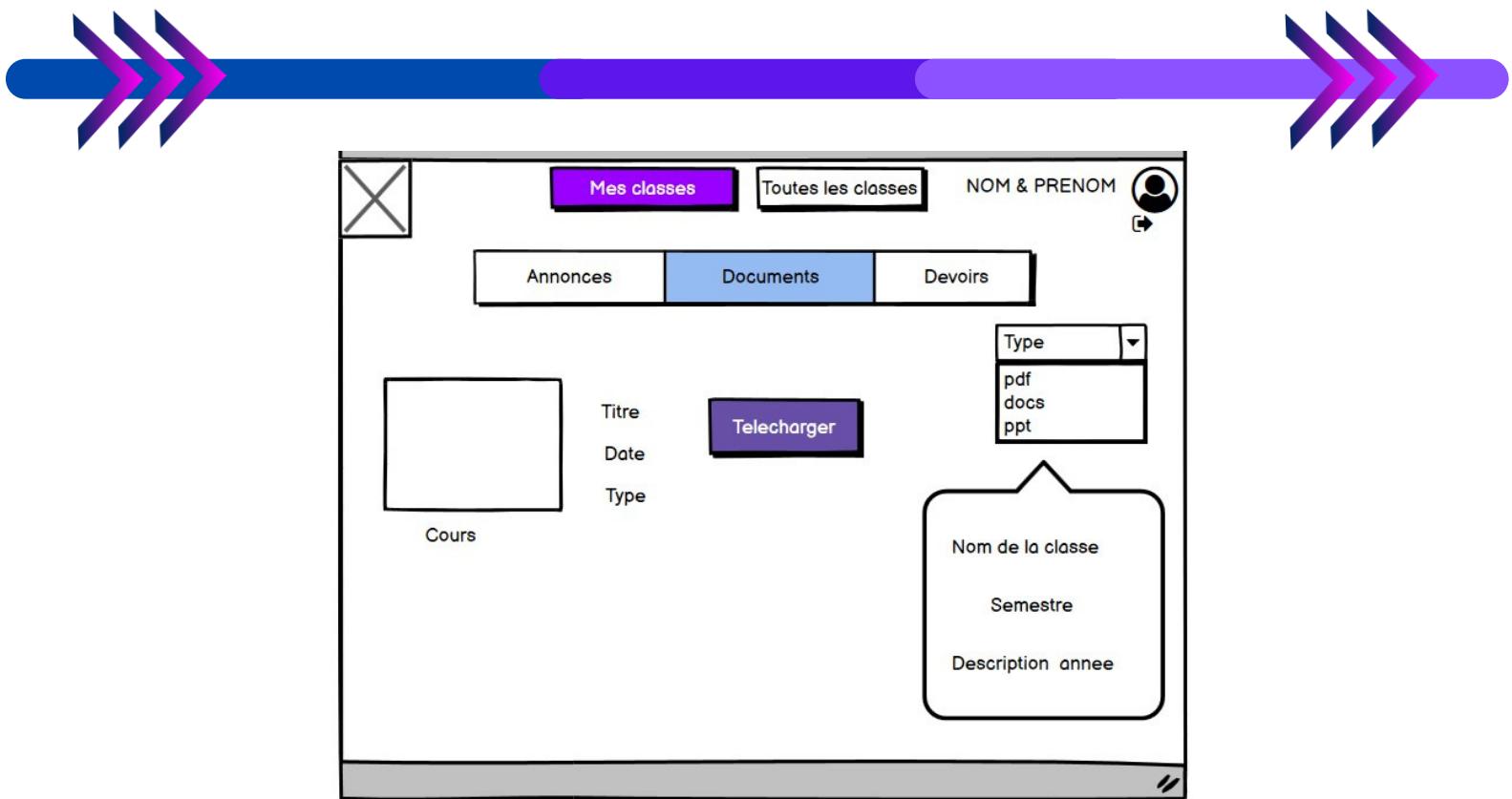


Figure 21 : Affichage des documents par type

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

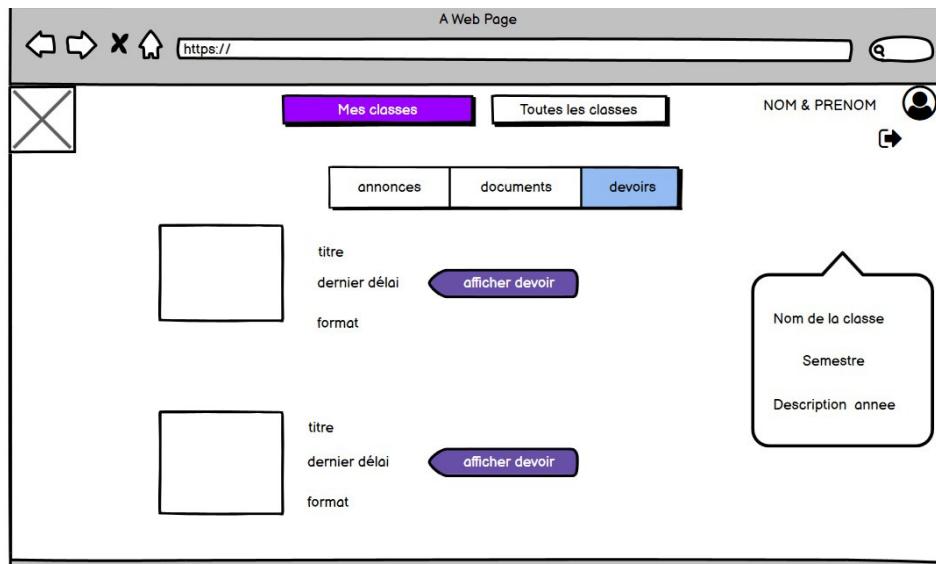
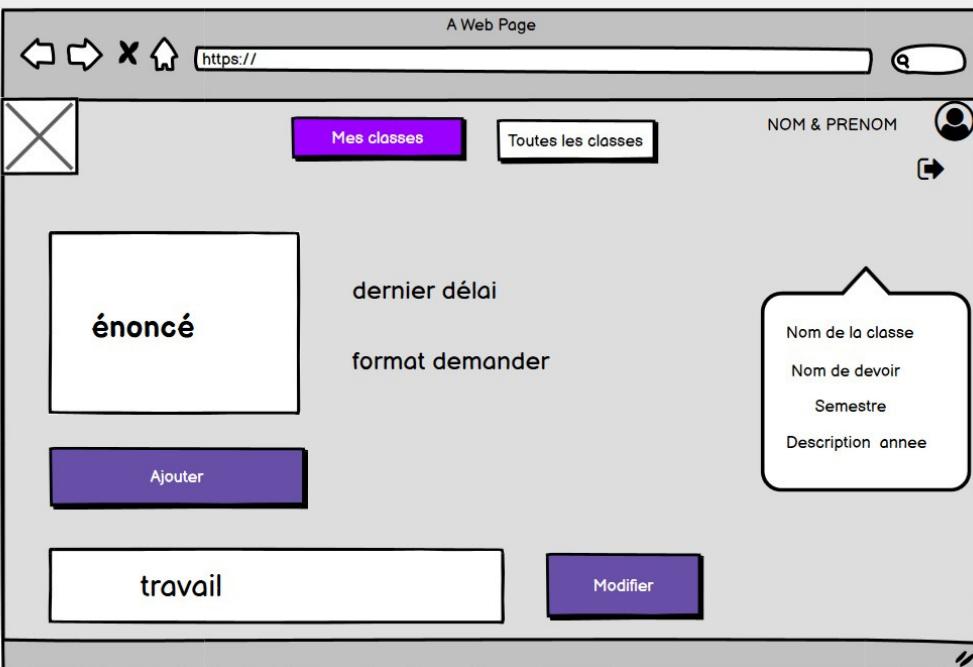


Figure 22: partie devoirs etudiant

Cette figure affiche les devoirs à faire



A zoomed-in view of the 'Ajouter votre travail' section. It features a large input field followed by a purple 'importer' button. Below this is a purple 'valider' button.

Figure 23: partie ajout des devoirs

Cette figure permet à l'étudiant de poster son travail s'il est dans les délais .sinon , il n'aura pas la possibilité d'ajouter ou modifier

L'espace enseignant:

A Web Page

https://

LOGO

Mes classes Annonce NOM & PRENOM

Créer une classe

Nom du cours

Nom de la formation

Semestre

Année universitaire

vider Creer

This screenshot shows a web page titled 'A Web Page' with a URL 'https://'. In the top right corner, there is a logo placeholder labeled 'LOGO', a user profile icon, and a sign-out button. The main navigation bar includes 'Mes classes' (highlighted in purple), 'Annonce', and 'NOM & PRENOM'. Below the navigation, there is a section titled 'Créer une classe' containing four input fields: 'Nom du cours', 'Nom de la formation', 'Semestre', and 'Année universitaire'. At the bottom of this section are two buttons: 'vider' (clear) and 'Creer' (create).

Figure 24: formulaire d'ajout d'une classe

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

A Web Page

https://

LOGO

Mes classes Annonce Document Devoir Absence Note Demande

Titre Description Date

Titre Description Date

Ajouter une annonce

Titre Date Description publier

This screenshot shows a web page titled 'A Web Page' with a URL 'https://'. In the top right corner, there is a logo placeholder labeled 'LOGO', a user profile icon, and a sign-out button. The main navigation bar includes 'Mes classes' (highlighted in purple), 'Annonce' (highlighted in blue), 'Document', 'Devoir', 'Absence', 'Note', and 'Demande'. Below the navigation, there are two sections, each containing 'Titre', 'Description', and 'Date' fields, followed by a delete icon. To the right, there is a 'Ajouter une annonce' section with similar fields and a 'publier' (publish) 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 .

Figure 28: partie absence

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

The screenshot shows a web application interface. At the top, there are navigation icons (back, forward, search) and a URL bar with 'https://'. Below the URL bar is a header with tabs: 'Mes classes' (purple), 'Annonce' (white), 'Annonce' (grey), 'Document' (grey), 'Devoir' (grey), 'Absence' (grey), 'Note' (blue, highlighted), and 'Demande' (grey). On the left, there is a logo icon. In the center, there is a table with student data:

Cne	Prenom	Nom	Note_devo	Note_norm	Note_rattrapag	Note_controle	Note_assiduit
A123456788	Aya	Moufid	20	20		20	20
V123245678	chaima	souhir					
R123512763	Kawtar	farrouj					

Below the table is a button 'Creer controle'. To the right, there is a section titled 'Coefficients' with five input fields:

- coff_Note_devoir: 0,25
- coff_Note_assiduite: 0,5
- coff_Note_normal: 0,75
- coff_Note_rattr: 0
- coff_Note_cont: 0

At the bottom, there is a 'Calculer note' button and a 'Note_reference' input field set to 15.

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).

The screenshot shows a web application interface. At the top, there are navigation icons (back, forward, search) and a URL bar with 'https://'. Below the URL bar is a header with tabs: 'Mes classes' (purple), 'Annonce' (white), 'Annonce' (grey), 'Document' (grey), 'Devoir' (grey), 'Absence' (grey), 'Note' (grey), and 'Demande' (blue, highlighted). On the left, there are two logo icons. In the center, there are two entries for student requests:

CNE	Nom prenom	Email	Date	Accepter	Refuser
				<input type="button" value="Accepter"/>	<input type="button" value="Refuser"/>
				<input type="button" value="Accepter"/>	<input type="button" value="Refuser"/>

Figure 30: partie demandes

Cette page permet à l'enseignant de 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 compréhensive sur les différentes cotés.

Toutes ces taches 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 :

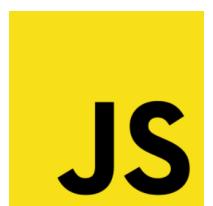
1. Environnement Hard :

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

2.LANGAGES 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.

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 tables 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 Wireframes

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 unifié 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.



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 de 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.

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

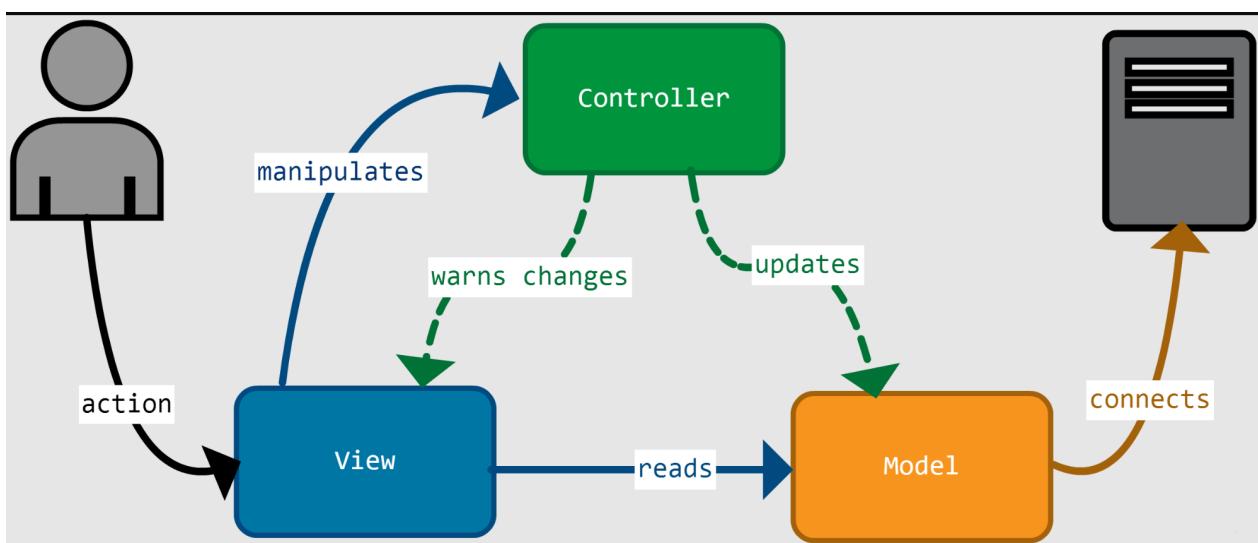


Figure 31: MVC

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 en commençant par le volet enseignant.



C

















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



