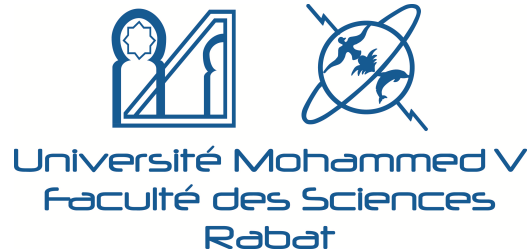


UNIVERSITÉ MOHAMMED V de Rabat
Faculté des Sciences



Département d'Informatique
Filière Licence fondamentale
en Sciences Mathématiques et Informatique

PROJET DE FIN D'ÉTUDES

intitulé :

Conception et réalisation d'un
Système d'information des étudiants
du département informatique

Gestion des inscriptions aux projets tutorés

Présenté par : ILYAS CHAOUA , MOHAMMED AMINE ACHALHI

soutenu le 4 Juin 2015 devant le Jury

M. El Mamoun SOUIDI	Professeur à la Faculté des Sciences - Rabat	<i>Président</i>
M. Oussama Mohamed REDA	Professeur à la Faculté des Sciences - Rabat	<i>Encadrant</i>
M. El Amrani Younès	Professeur à la Faculté des Sciences - Rabat	<i>Examineur</i>

Année universitaire 2014-2015

“ Mesurer la progression du développement d’un logiciel à l’aune de ses lignes de code revient à mesurer la progression de la construction d’un avion à l’aune de son poids. ’ ”
Bill Gates.

Remerciements

À la remise du présent rapport, nous tenons non seulement par nécessité, mais par respect et gratitude d'exprimer nos vifs et sincères remerciements à Mr OUSSAMA REDA notre encadrant de nous avoir encadré tout au long de ce projet

Nous adressons nos fidèles remerciements à nos honorables membres du jury d'avoir accepté de consacrer leur précieux temps à l'évaluation de ce travail. Ainsi que tout le corps professoral de la faculté de sciences pour leurs qualités d'enseignement.

En fin on remercie tous ceux qui ont participé de près ou de loin à l'élaboration de ce projet.

Résumé :

Ce projet vise à développer un système d'information des étudiants du département informatique. Mais, pour aboutir à cette fin, nous allons tout d'abord effectué une étude conceptuelle de l'application. Cette dernière nous permettra, en effet, d'accéder facilement à la réalisation de l'application en organisant les idées et en structurant le processus de codage suivant des diagrammes. L'application a été implémenté par diverses technologies en se basant sur l'étude conceptuelle. Le système de gestion de base de données choisi fut MySQL. L'application a été implémenté avec Laravel5 et Bootstrap3, qui sont des frameworks permettant de créer rapidement et efficacement un site web complexe et flexible.

Mots clés : Système d'information des étudiants, Laravel5, UML, Bootstrap3, MySQL, Ajax

Abstract :

Today, the major concern of universities is the difficulty of communication between administration, student and teacher which affects the quality of education. This graduation project aims to design and implement an information system for the student of mathematics and computer science department of the faculty of sciences Rabat. We tried all along our work to build our increment by increment application using the Scrum methodology and based on the Gantt chart that form good software engineering practices.

The conceptual study was made by Unified Modeling Language to provide a standard way to visualize the design of a system. The use case diagram that describes the functionality of the system, the class diagram that describes the types of objects that make up a system, the sequence diagrams to represent the interactions between objects identifying the exchanges chronology.

A detailed study of technologies, frameworks and applications architectures was done to achieve a better solution. The database management system chosen was MySQL. And then we implemented the application using Laravel5, which is a PHP framework that allows to quickly and efficiently create a complex website based in web server, then Bootstrap3 to build web pages on the client side.

Keywords : Students information system, Laravel5, UML, Bootstrap3, MySQL, Ajax

Table des matières

1	Introduction générale	3
2	Cahier de charge et spécifications des besoins	5
2.1	Présentation du projet :	6
2.1.1	Objectif :	6
2.1.2	Cible et caractéristique :	6
2.1.3	Description du contenu :	6
2.1.4	Études de l'existant :	6
2.1.5	Spécification des besoins fonctionnels :	6
2.1.6	Spécification des besoins non fonctionnels :	8
2.1.7	Méthodes de gestion de projet : SCRUM et Diagramme de Gantt . .	8
2.2	Conclusion	12
3	Conception fonctionnelle	13
3.1	Introduction	14
3.2	Vue fonctionnelle du système	14
3.2.1	Acteurs	14
3.2.2	Diagramme de cas d'utilisation	14
3.2.3	Description textuelle des cas d'utilisation	16
3.3	Vue statique du système : Diagramme de classe	19
3.4	Vue dynamique du système	20
3.4.1	Diagrammes de séquences	20
3.5	Modélisation conceptuelle et logique des données	24
3.5.1	Définition	24
3.5.2	Modèle Conceptuel des données :	24
3.5.3	Modèle logique des données	24
3.6	Conclusion	27
4	Conception technique	29
4.1	Introduction	29
4.2	Environnement et technologies logicielles	29
4.2.1	Technologies utilisées	29
4.3	Framework	30
4.3.1	Définition	30

4.3.2	Exemple de Framework web :	31
4.3.3	Architecture logicielle	31
4.4	Choix technologiques	32
4.5	Solution Retenue	34
4.6	Conclusion	35
5	Réalisation	37
5.1	Introduction	38
5.2	Cas : S'inscrire au système.	38
5.3	Cas : Suivre des cours.	38
5.4	Cas : Visualiser le Graphe de notes.	39
5.5	Cas : Visualiser les Statistiques des absences	39
5.6	Cas : Télécharger des fichiers	40
5.7	Cas : Demander des services	40
5.8	Cas : S'inscrire au module de PFE	41
5.9	Cas : Visualiser les emplois du temps.	41
5.10	Cas : Affecter des PFE.	42
5.11	Cas : Ajouter des fichiers.	42
5.12	Conclusion	42
6	Conclusion et perspectives	43
	Bibliographie	45

Table des figures

2.1.1 Méthode de développement Agile : Scrum	9
2.1.2 La listes des taches à réaliser	10
2.1.3 Diagramme de Gantt correspond à notre projet	11
3.2.1 Diagramme de cas d'utilisation	15
3.3.1 Diagramme de classe	19
3.4.1 Diagramme de séquence du cas "S'inscrire"	20
3.4.2 Diagramme de séquence du cas "Demander service".	21
3.4.3 Diagramme de séquence du cas "Ajouter Cour".	22
3.4.4 Diagramme de séquence du cas "S'inscrire en module PFE".	23
3.5.1 Modèle conceptuelle de la base de données	25
4.4.1 Statistiques de popularité Framework PHP fin 2013	33
4.4.2 Template AdminLTE	34
4.5.1 Figure Fonctionnement de l'application.	35
5.2.1 Page d'inscription dans le système.	38
5.3.1 Page d'accueil.	38
5.4.1 Page des notes par semestre de l'étudiant.	39
5.5.1 Page d'absences par module de l'étudiant.	39
5.6.1 Page des fichiers.	40
5.7.1 Page des services.	40
5.8.1 Page d'inscription au module de PFE.	41
5.9.1 Page des emplois du temps.	41
5.10. Page d'affectation par l'administrateur.	42
5.11. Page d'impression des demandes par l'administrateur.	42

Chapitre 1

Introduction générale

Dans le cadre de notre formation en licences fondamentales au sein de la faculté de sciences de Rabat, l'établissement cherche à faire évoluer les compétences de ses étudiants par divers moyens tels que les travaux pratiques, les projets et particulièrement les projets tutorés, pour cela, nous avons été amené à réaliser un projet de fin d'études qui a pour thème " Conception et réalisation d'un Système d'informations pour les étudiants du département informatique " afin de concrétiser nos acquis durant notre formation.

Le département cherche à améliorer la gestion et la communication avec ses étudiants, en mettant en place un nouveau système d'information pour les étudiants. Ce dernier permettra de collecter, stocker, traiter et diffuser de l'information concernant des étudiants. Il a pour but d'échanger des informations entre les étudiants et le corps professoral.

Une présence d'un tel système est indispensable pour le département afin de faciliter la communication avec les étudiants, optimiser le processus de gestion, la réduction de certains coûts et l'amélioration de certains services

Chapitre 2

Cahier de charge et spécifications des besoins

2.1 Présentation du projet :

2.1.1 Objectif :

L'objectif principal de ce projet est de développer une application web qui permet de gérer le système d'information des étudiants du département informatique. Pour aboutir à cette fin, nous allons tout d'abord effectué une étude conceptuelle de l'application. Cette dernière nous permettra, en effet, d'accéder facilement à la réalisation de l'application en organisant les idées et en structurant le processus de codage suivant des diagrammes.

2.1.2 Cible et caractéristique :

Description de la cible : Étudiants (SMI), administrateurs, enseignants.

Le projet vise principalement les étudiants du département informatique.

- Cible principale et description : les Étudiants du département.
- Cible secondaire et description : les enseignants et les administrateurs.

2.1.3 Description du contenu :

La première page c'est une page d'authentification qui contiendra un formulaire d'authentification pour les étudiants déjà inscrits et un onglet registre qui redirige l'internaute vers un formulaire pour l'inscription, la vérification sera faite par la suite, après l'authentification l'étudiant et redirigé vers la page principale, elle contiendra par ailleurs un sidebar à travers lequel on peut effectuer les fonctionnalités que propose le système.

2.1.4 Études de l'existant :

Le système existant, s'appuie sur la vérification manuelle des données, ce qui éventuellement entraine plusieurs anomalies :

- Difficulté d'accès aux données ;
- Difficulté de la communication entre administration et étudiant ;
- Gaspillage de temps ;
- Limitation des services estudiantins ;
- Absence d'ergonomie.

2.1.5 Spécification des besoins fonctionnels :

L'application que nous allons développer devra regrouper toutes les fonctionnalités nécessaires pour :

- S'authentifier.
- S'inscrire.
- Visualiser des cours.
- Rechercher des cours.
- Visualiser les statistiques des notes.
- Visualiser les statistiques des absences.
- Ajouter des cours.

- Visualiser les emplois du temps.
- Demander un service.
- S’inscrire dans un pfe.

S’authentifier :

L’utilisateur devra pouvoir s’authentifier à travers un email et un mot de passe. Le système vérifie l’authentification et le type d’internaute (étudiant, enseignant, administrateur ou un simple internaute).

S’inscrire :

Chaque utilisateur pourra effectuer une inscription en saisissant ses informations personnelles et en choisissant son type (Étudiant / Enseignant), afin de permettre au système de vérifier son existences au sein du département.

Visualiser des cours :

Chaque étudiant pourra consulter la liste des cours à tout moment et les télécharger.

Rechercher cours par mot-clé :

L’étudiant devra pouvoir trouver le plus rapidement possible un cours recherché par un mot clé qui lui est associé.

Visualiser les statistiques des notes :

L’étudiant devra pouvoir visualiser les notes de chaque module selon les semestres.

Visualiser les statistiques des absences :

L’étudiant devra pouvoir visualiser le pourcentage de ses absences dans chaque module.

Ajouter des cours :

L’administrateur devra ajouter un cours et lui associer un mot clé. Ce dernier sera automatiquement affiché chez l’étudiant.

Visualiser les emplois du temps :

Chaque internaute pourra visualiser et télécharger l’ensemble des emplois du temps du département informatique.

Demander un service :

Chaque étudiant devra pouvoir effectuer une demande d'un service auprès de l'administration, cette demande sera traitée par l'administrateur qui recevra la liste des étudiants ayant demandé un service.

S'inscrire dans un pfe :

Chaque étudiant devra remplir une fiche contenant les informations sur les deux binômes d'un pfe. Un administrateur sera chargé d'affecter un encadrant de pfe à chaque binôme.

2.1.6 Spécification des besoins non fonctionnels :

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt ils identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application sont les suivants :

- Sécurité : Les comptes des utilisateurs sont sécurisés par mot de passe (longueur, code système, expiration de sessions, etc)
- Fiabilité : Bon fonctionnement de l'application sans détection de défaillance.
- Performance : L'application répond à toutes les exigences des internautes d'une manière optimale.
- Convivialité : Un design graphique clair et simple pour faciliter l'utilisation à l'utilisateur.
- Portabilité : L'application est multiplateforme. Elle fonctionne sur tout les systèmes d'exploitation et tout type de terminal.

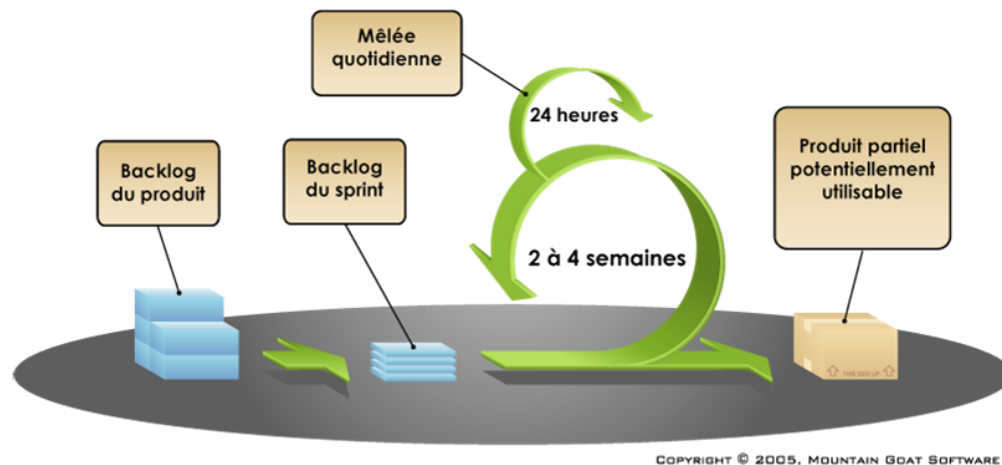
2.1.7 Méthodes de gestion de projet : SCRUM et Diagramme de Gannt

Scrum

Scrum est un cadre de travail pour le développement et la maintenance de produits complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. Cette définition se compose des rôles, des événements et des artéfacts de Scrum, ainsi que des règles qui les lient. La version originale du Guide Scrum (Scrum Guide) est l'œuvre de Ken Schwaber et Jeff Sutherland, les créateurs de Scrum. Le principe de base de Scrum est le suivant :

- Premièrement, spécifier le maximum des fonctionnalités à réaliser pour former le Backlog du produit.
- En second lieu définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération.
- Par la suite focaliser l'équipe de façon itérative sur l'ensemble de fonctionnalités à réaliser, dans des itérations appelées Sprints
- Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel appelé incrément.

S'il est difficile de définir la nature de Scrum, sa mise en place est beaucoup plus simple et peut être résumée par la figure suivante : Le choix de Scrum comme une méthodologie de



Méthode de développement Agile : Scrum

pilotage et de développement logiciel agile itératif qui s'est basé sur les avantages suivants :

- Plus de créativité et de souplesse.
- Capacité d'adaptation au changement grâce à des itérations courtes,
- Flexible dans la mesure où une équipe de développement fonctionne comme une unité, pour atteindre un objectif commun".

Vu que Scrum ne couvrant que les aspects de gestion de projet, et pour compléter le vide laissé en matière de pratiques de développement, nous avons pris la décision de coupler Scrum avec une autre méthodologie de gestion est celle du Diagramme de Gantt et qui couvre les bonnes pratiques d'ingénierie logicielle notamment le développement dirigé par le test et la programmation en binôme. voir la référence[12]

Diagramme de Gantt

Un diagramme de Gantt est un type de graphique à barres qui illustre un calendrier de projet, c'est à dire le début et de fin dates des éléments terminaux et des éléments de synthèse d'un projet. Les éléments terminaux sont des éléments de synthèse qui constituent la structure de la répartition du travail dans le projet. Les diagrammes de Gantt modernes montrent également la dépendance des relations entre les activités. Bien que maintenant considéré comme une technique de gestion, diagrammes de Gantt ont été considérés comme révolutionnaire lors de son lancement. Il est également utilisé dans la technologie de l'information pour représenter les données qui ont été recueillies. Le diagramme de Gantt nous permettra de mieux visualiser les étapes de notre travail ainsi que d'identifier les étapes conditionnelles aux suivantes et celles à démarrer en priorité et différencier les tâches.

La mise en place peut être résumée par ces deux figures réaliser par l'application en ligne "Smart Sheet" :

Lors du pilotage d'un projet, l'identification des risques critiques présente une étape indispensable pour la réussite de ce dernier. Pour notre cas, le seul risque qui peut nous ralentir est lié la complexité de l'application et aux différentes contraintes à respecter. [13]

Diagramme de Gantt

	At Risk	Task Name	Start Date	End Date	Assigned To	% Complete	Duration	Predecessors
1		Need Help? Learn more about this template.						
2								
3		Du 20 Mars à 22 Juin 2015	03/20/15	06/22/15			67	
4		Spécification des besoins fonctionnels	03/20/15	04/14/15		100%	18	
5		Etude de l'existant	03/20/15	03/24/15	Ilyas et amine	100%	3	
6		Use case	03/25/15	03/31/15	Ilyas et amine	100%	5	
7		Diagramme de classes	04/01/15	04/07/15	Ilyas et amine	100%	5	6
8		Traduction du modèle conceptuelle de la base de données	04/08/15	04/14/15	Ilyas et amine	100%	5	7
9		Dessin de la maquette	04/08/15	04/08/15	Ilyas	100%	1	
10		Diagramme de séquence	04/01/15	04/06/15	Amine	100%	4	6
11		Conception technique	04/15/15	04/17/15		100%	3	
12		Choix du technologies	04/15/15	04/15/15	Amine et Ilyas	100%	1	4
13		Etudes des frameworks	04/16/15	04/17/15	Amine et Ilyas	100%	2	12
14		Réalisation	04/20/15	06/30/15		100%	52	
15		Fonctionnalités Inscription et Authentification et Mailing	04/20/15	04/29/15	Ilyas et Amine	100%	8	11
16		Intégration du modèle Admin LTE	04/30/15	05/01/15	Ilyas	100%	2	15
17		Fonctionnalités Services	05/04/15	05/06/15	Amine	100%	3	16
18		Fonctionnalité Gestion du profile	05/07/15	05/08/15	Ilyas	100%	2	17
19		Fonctionnalité Rechercher	05/11/15	05/15/15	Amine	100%	5	18
20		Fonctionnalités Progression et Absences	05/18/15	05/29/15	Ilyas et Amine	100%	10	19
21		Fonctionnalité d'inscription en PFE	06/01/15	06/05/15	Ilyas	100%	5	20
22		Réalisation de la page d'administrateur du pfe	06/01/15	06/16/15	Ilyas	100%	12	20
23		Réalisation de la page d'administrateur des services	06/01/15	06/16/15	Amine	100%	12	20
24		Tests unitaires et maintenances	06/17/15	06/30/15	Ilyas et Amine	100%	10	23

La listes des taches à réaliser

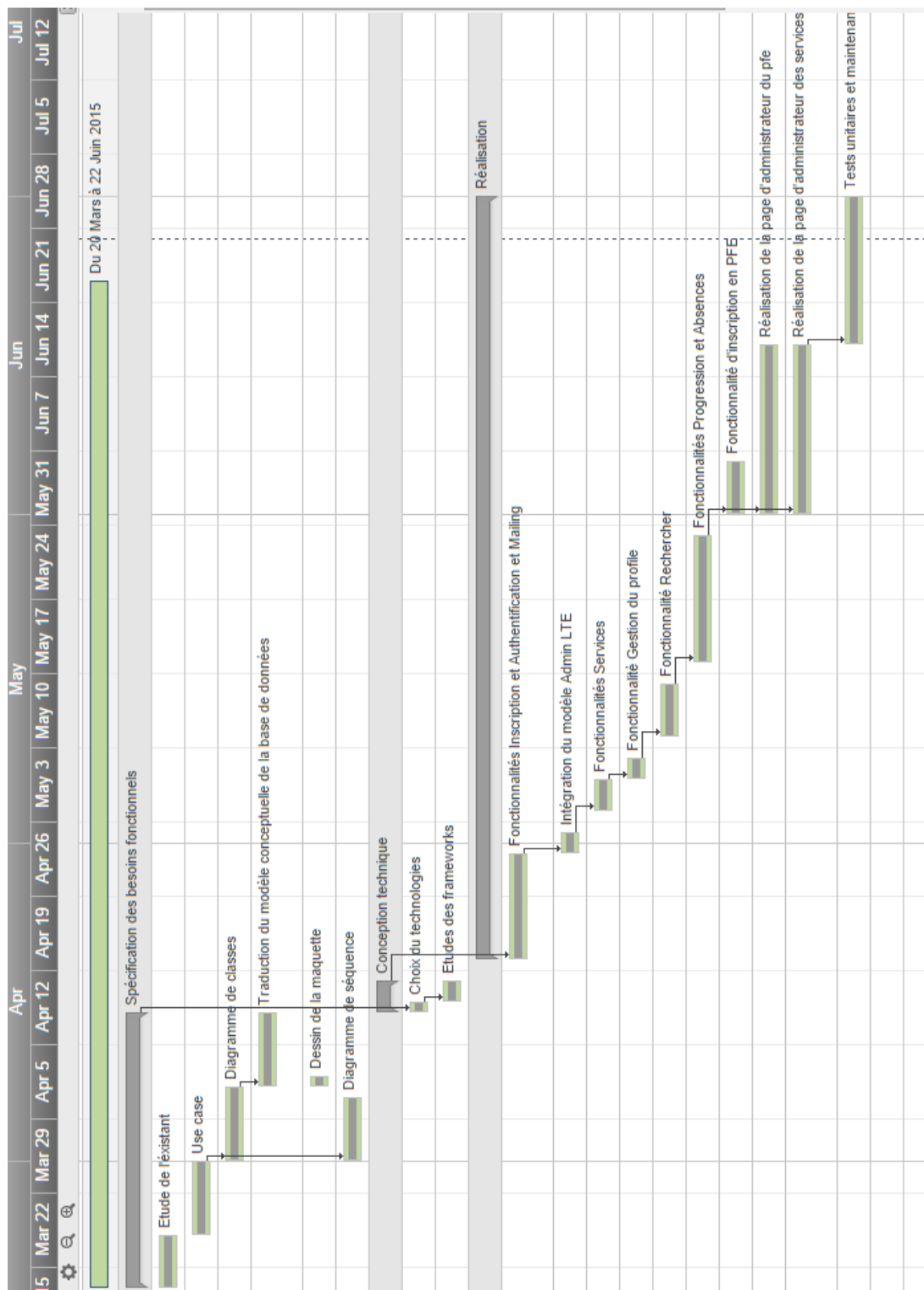


Diagramme de Gantt correspond à notre projet

2.2 Conclusion

Dans ce chapitre nous avons présentés le cadre général de notre projet en déterminant la problématique et les périmètres tout en proposant une solution envisagée pour faire face à la situation courante. Nous avons dévoilé les exigences des besoins fonctionnels et non fonctionnels. Comme nous avons choisi d'adopter la méthodologie Scrum pour la conception de notre futur système accompagné d'un diagramme de Gantt. Dans le chapitre suivant nous allons reproduire les différents besoins cités précédemment sous forme de diagrammes UML.

Chapitre 3

Conception fonctionnelle

3.1 Introduction

Dans cette section, nous allons reproduire les différents besoins cités précédemment sous la forme de diagrammes UML.

UML est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la conception orientée objet [3].

Au final, le langage UML est une synthèse de tous les concepts et formalismes méthodologiques les plus utilisés, pouvant être utilisés, grâce à sa simplicité et à son universalité, comme langage de modélisation pour la plupart des systèmes devant être développés.

3.2 Vue fonctionnelle du système

3.2.1 Acteurs

Un acteur est un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Les acteurs qui peuvent interagir avec notre application sont :

1. Étudiant du département informatique, c'est l'acteur le plus important, celui pour lequel le système existe.
2. Administrateur des projets de fin des études il a pour rôle d'affecter des encadrant aux binômes.
3. Administrateur des services il a pour rôle de suivre le processus des demandes administratives et ajouter des cours.

3.2.2 Diagramme de cas d'utilisation

Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système)[3]. Il représente toutes les fonctionnalités que le système doit fournir.

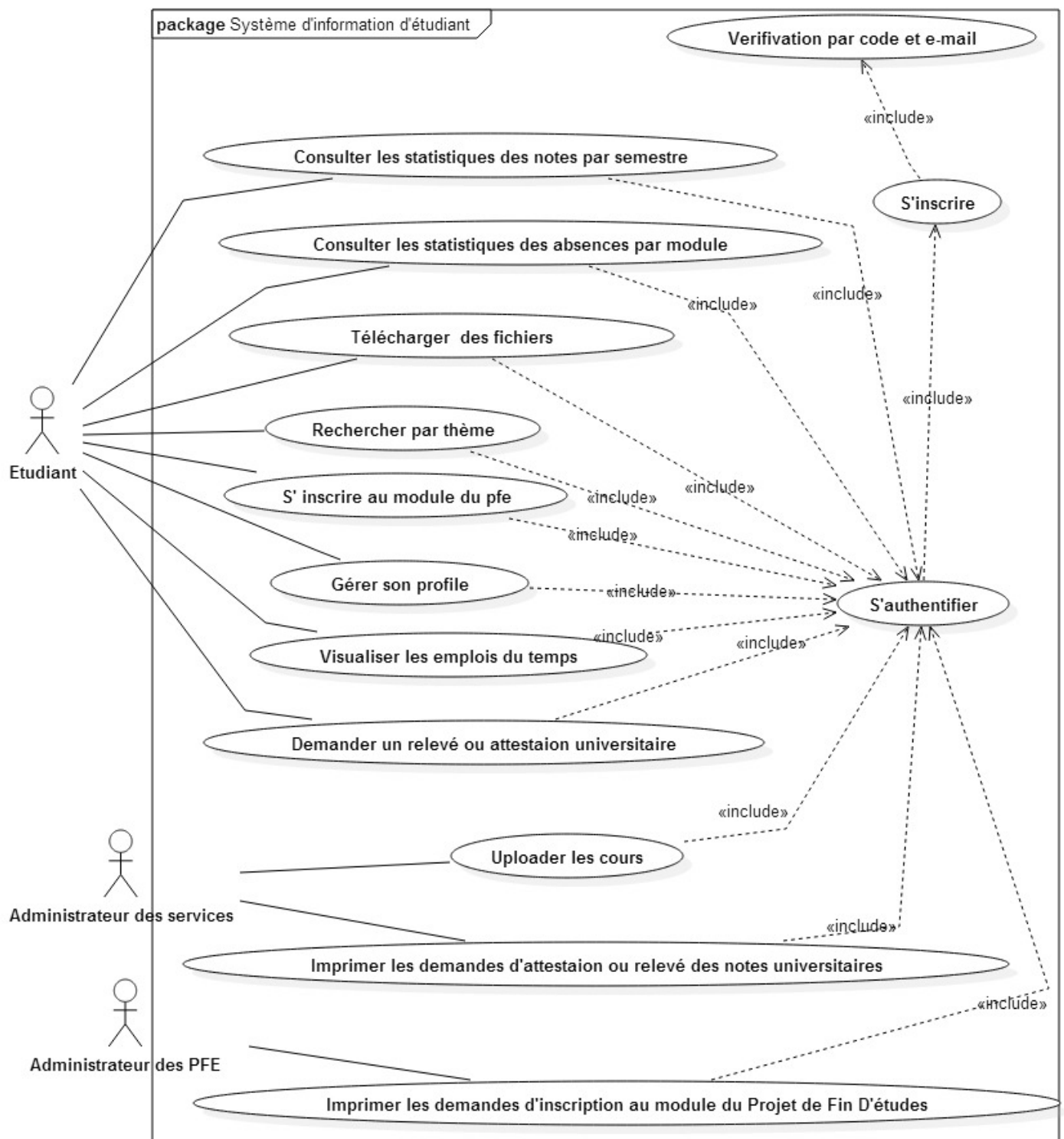


Diagramme de cas d'utilisation

3.2.3 Description textuelle des cas d'utilisation

Cas "S'inscrire"

1. Acteurs principaux : Étudiant
2. Objectifs : Ce cas d'utilisation vise à décrire toutes les étapes relatives à l'inscription d'un internaute afin d'effectuer les différentes opérations correspondantes à un étudiant.
3. Pré-conditions :
 - Disponibilité d'accès au réseau Internet.
 - Serveur accessible 24 h/24 h et 7j/7j .
 - Navigateur en bon état de fonctionnement.
4. Postconditions : Utilisateur inscrit et authentifié par une adresse électronique et un mot de passe.
5. Scénario nominal :
 - L'internaute demande à s'inscrire.
 - Le système demande à l'internaute de choisir le type du compte à créer.
 - L'internaute choisit le type du compte.
 - Le système affiche le formulaire d'inscription.
 - L'internaute remplit les champs du formulaire puis il valide.
 - Le système vérifie si un utilisateur n'est pas déjà inscrit avec la même adresse email et qu'il fait bien partie du département informatique puis, enregistre le nouveau compte et envoie un email de confirmation à l'internaute.
6. Alternatifs :
 - Lors de la vérification, si le système trouve que l'internaute est déjà inscrit,il sera redirigé,alors, vers la page d'authentification .

Cas "Conculter les statistiques des notes par semestre "

1. Acteurs principaux : Étudiant
2. Objectifs : Ce cas d'utilisation vise à afficher un graphe de moyennes des notes par semestre décrivant ainsi la progression de l'étudiant au fil du temps.
3. Pré-conditions :
 - Être authentifié par le système.
 - La table des notes et leurs coefficients est remplie.
4. Postconditions : néant
5. Scénario nominal :
 - L'étudiant demande la page des notes.
 - Le système récupère les différentes notes reliés à l'étudiant et calcule les moyennes de chaque semestre.
 - Le système de vue trace le graphe en se basant sur ces moyennes.
 - Le système affiche le résultat.
6. Alternatifs : si L'étudiant n'a aucune note dans la table des évaluations,le graphe sera vide.

Cas "Consulter les statistiques des absences par module "

1. Acteurs principaux : Étudiant
2. Objectifs : Ce cas d'utilisation a pour but d'afficher des graphes décrivant ainsi le pourcentage d'absence pour chaque module.
3. Pré-conditions :
 - Être authentifié par le système.
 - Être inscrit au moins dans un module.
4. Postconditions : néant
5. Scénario nominal :
 - L'étudiant demande la page des absences.
 - Le système récupère les différentes absences reliée à l'étudiant et calcule le pourcentage pour chaque module.
 - Le système de vue trace les graphes.
 - Le système affiche les résultats.
6. Alternatifs : Si l'étudiant n'a aucune absence, la page de statistique sera vide.

Cas "S'inscrire dans le module du Projets tutorés "

1. Acteurs principaux : Étudiant
2. Objectifs : Ce cas d'utilisation vise à décrire toutes les étapes relatives à l'inscription des étudiants au module projet de fin d'études.
3. Pré-conditions :
 - Être authentifié par le système.
 - Être inscrit dans le dernier semestre.
 - Avoir choisi son binôme.
4. Postconditions : l'étudiant imprime le formulaire d'inscription et le dépose chez le secrétariat
5. Scénario nominal :
 - L'étudiant demande à s'inscrire au module du PFE.
 - Le système affiche le formulaire d'inscription.
 - L'étudiant remplit les champs du formulaire puis il valide.
 - Le système vérifie si le binôme fait bien partie du département informatique puis enregistre la nouvelle demande et renvoie un fichier à imprimer.
6. Alternatifs :
 - si le binôme est déjà inscrit dans ce module, une page d'erreurs s'affichera.

Cas "Demander un relevé ou attestation universitaire"

1. Acteurs principaux : Étudiant
2. Objectifs : Ce cas d'utilisation décrit les étapes de la demande d'un relevé de notes ou attestation universitaire.

3. Pré-conditions :
 - Être authentifié par le système
4. Postconditions : l'étudiant doit de se présenter chez l'administration pour récupérer sa demande.
5. Scénario nominal :
 - L'étudiant demande un service.
 - Le système affiche le formulaire des services.
 - L'étudiant remplit les champs du formulaire puis il valide.
 - Le système enregistre la nouvelle demande.
6. Alternatifs : néant

Cas "Imprimer les demandes d'attestation ou relevé de notes"

1. Acteurs principaux : Administrateur des services
2. Objectifs : Ce cas d'utilisation décrit les étapes d'impression des relevés de notes ou des attestations universitaires.
3. Pré-conditions :
 - Être authentifié par le système .
4. Postconditions : l'administrateur doit présenter le fichier des demandes à l'administration.
5. Scénario nominal :
 - L'administrateur des services reçoit les demandes de la semaine courantes.
 - Le système affiche un tableau des demandes.
 - L'administrateur valide la demande.
 - Le système génère un fichier d'impression.
6. Alternatifs : Si aucune demande n'est enregistrée dans la semaine courante, rien ne sera imprimé.

Cas "Imprimer les demandes d'inscription au module du projet de fin d'études"

1. Acteurs principaux : Administrateur des PFE
2. Objectifs : Permettre à l'administrateur des PFE d'affecter un enseignant pour chaque binôme.
3. Pré-conditions :
 - Être authentifié par le système.
 - Tous les étudiants doivent confirmer l'inscription.
4. Postconditions : néant
 - L'administrateur des services demande les binômes inscrits.
 - Le système affiche un tableau des binômes.
 - L'administrateur affecte des enseignants aux binômes.
 - Le système génère un fichier d'impression.
5. Alternatifs : néant

3.3 Vue statique du système : Diagramme de classe

Le diagramme de classes exprime la structure statique du système en terme de classes et de relations entre ces classes. L'intérêt du diagramme de classes est de modéliser les entités du système d'information[3]. Ces informations sont regroupées ensuite dans des classes 3.3.1.

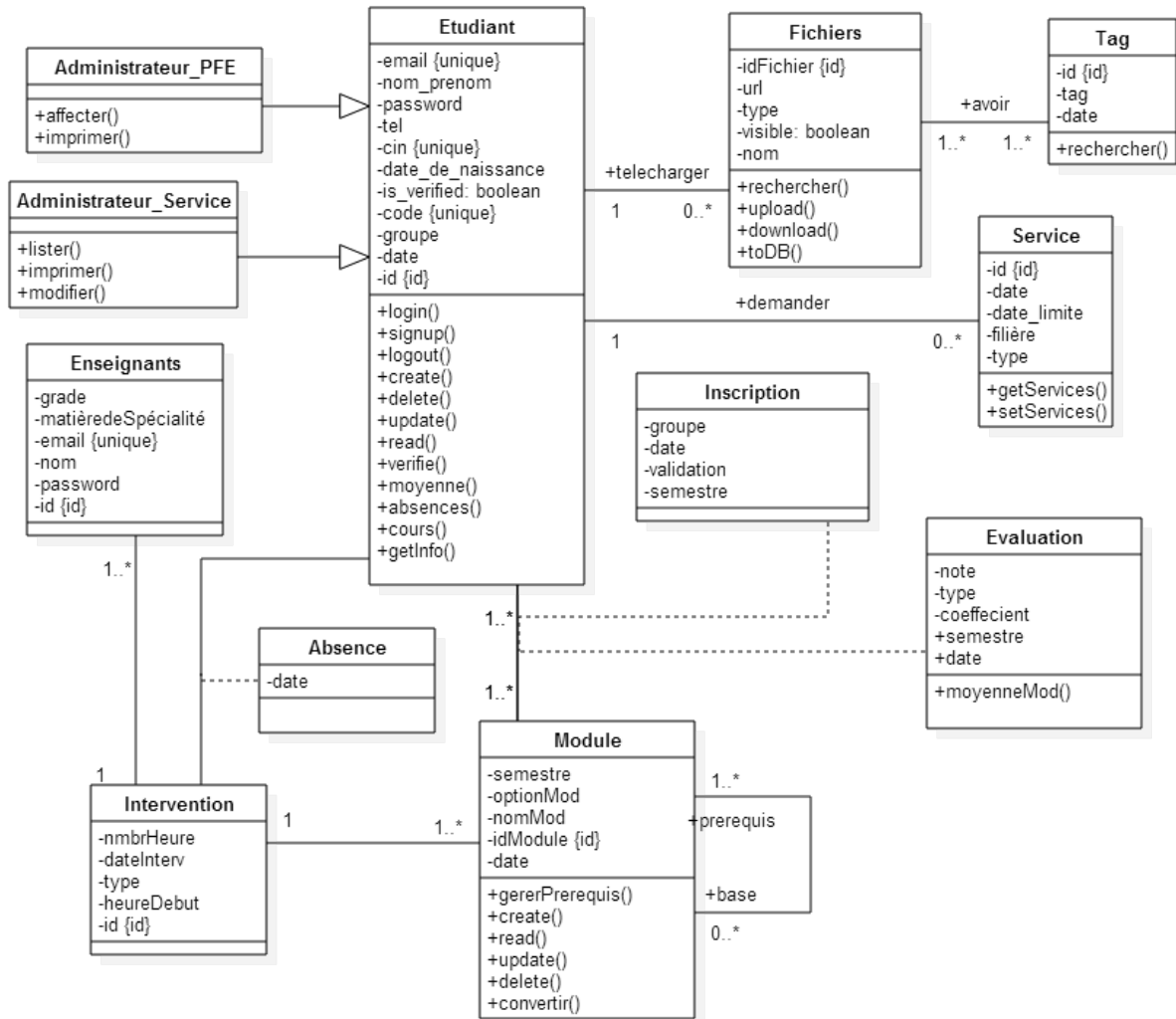


Diagramme de classe

Le diagramme de Classe représente d'une manière abstraite des objets du système qui vont interagir ensemble pour la réalisation des cas d'utilisation, sans prendre en compte le facteur du temps.

3.4 Vue dynamique du système

3.4.1 Diagrammes de séquences

Pour schématiser la vue comportementale de notre système informatique, nous faisons recours au diagramme de séquence d'UML. Ce diagramme permet de présenter les interactions entre l'acteur et le système avec des messages présentés dans un ordre chronologique. Le diagramme de séquence système traite le système informatique comme étant une boîte noire. Le comportement du système est décrit de l'extérieur sans avoir d'idée la réalisation. Nous pouvons, alors, constater que certains cas d'utilisations sont similaires, c'est pour cette raison que nous avons choisi de traiter quelques exemples.

Cas "S'inscrire "

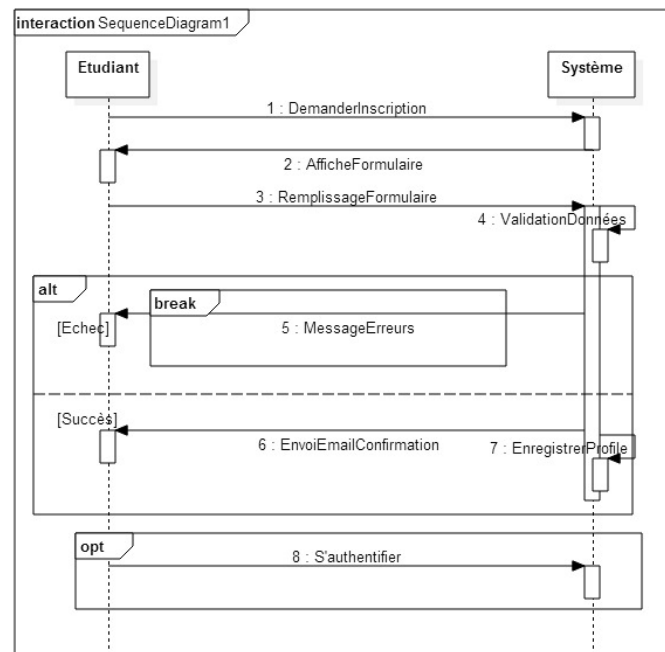


Diagramme de séquence du cas "S'inscrire"

- L'acteur principal, étudiant, est représenté à gauche du diagramme et le système à droite.
- L'étudiant demande au système le formulaire de création de compte.
 - Le système lui affiche le formulaire.
 - L'étudiant remplit et valide le formulaire.
 - Si les informations que l'étudiant a saisies sont invalides : le système génère une page d'erreur et le cas d'utilisation se termine en échec. Sinon : le système enregistre les informations dans la base de données et lui envoie un email de vérification.
 - L'étudiant pourra se connecter

Cas "Demander un relevé ou attestation universitaire"

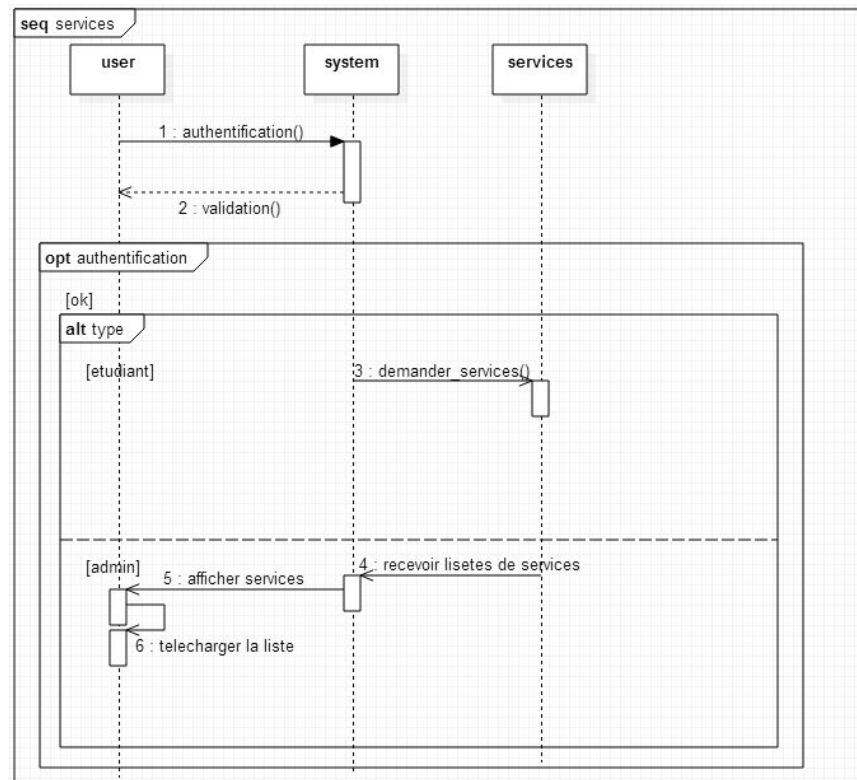


Diagramme de séquence du cas "Demander service".

L'acteur principal, étudiant, est représenté à gauche du diagramme et le système à droite.

- L'utilisateur commence d'abord par s'authentifier
- Si l'authentification est validé et l'utilisateur est un étudiant il pourra demander des services
- Sinon Si l'authentification est validé et l'utilisateur est un administrateur il recevra une liste des étudiants qui ont demandé un service particulier, la liste sera affiché et téléchargée par la suite.

Cas "Ajouter cour"

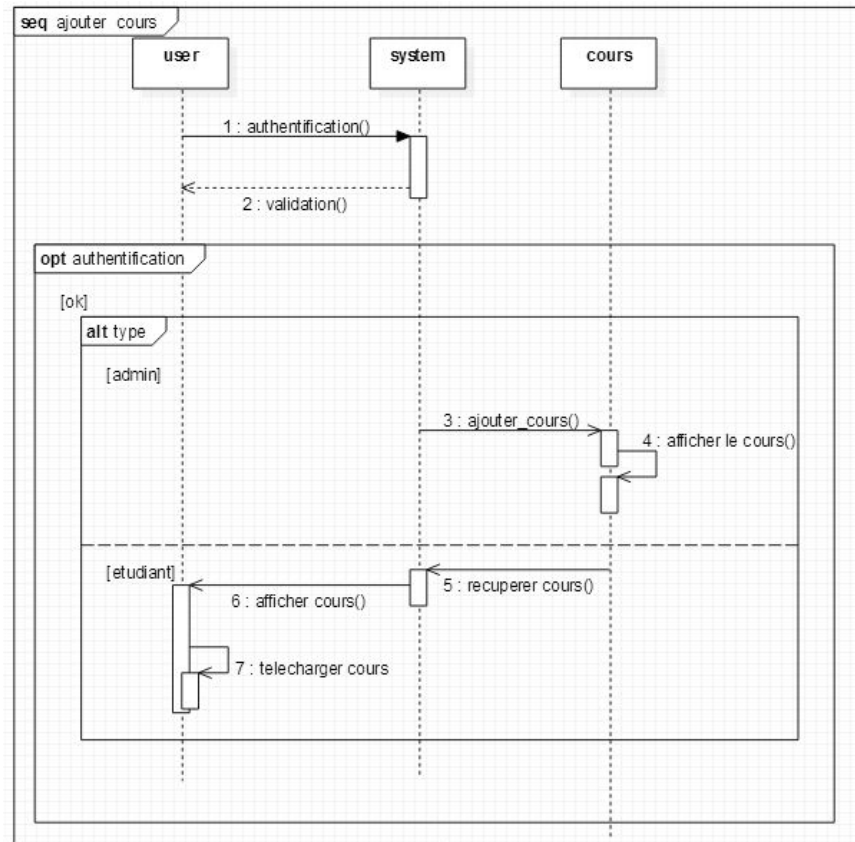


Diagramme de séquence du cas "Ajouter Cour".

L'acteur principal, Administrateur des services, est représenté à gauche du diagramme et le système à droite.

- L'utilisateur commence d'abord par s'authentifier
- Si l'authentification est validé et l'utilisateur est un administrateur il pourra ajouter un cours et l'afficher
- Sinon si l'authentification est validé et l'utilisateur est un étudiant il pourra voir la liste des cours et les télécharger.

Cas "S'inscrire en module PFE"

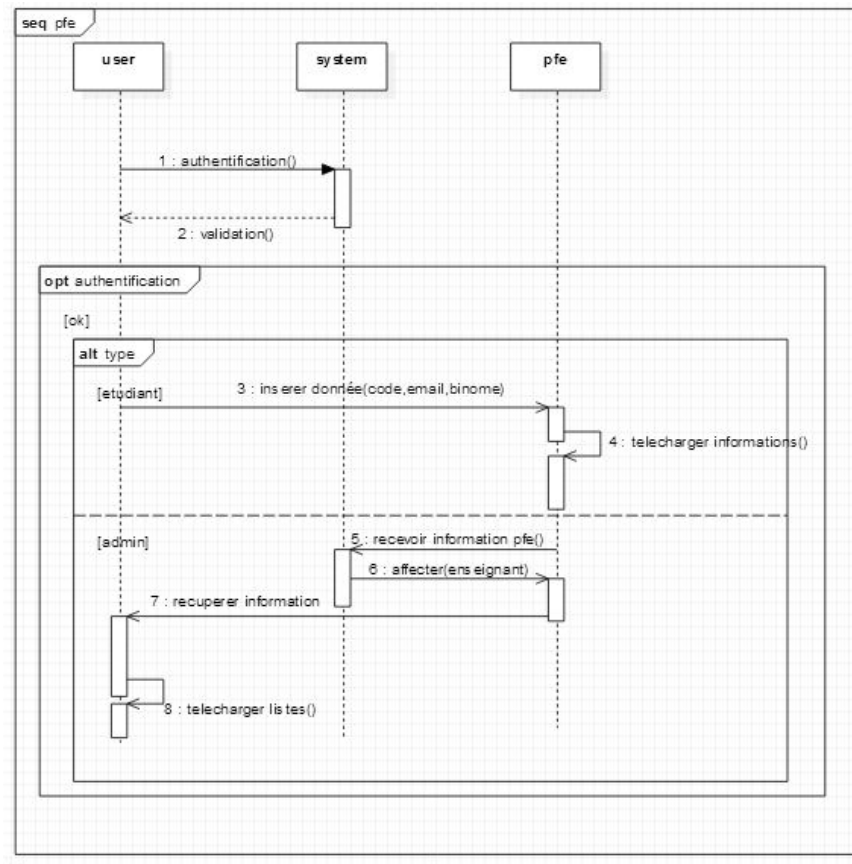


Diagramme de séquence du cas "S'inscrire en module PFE".

L'acteur principal, étudiant, est représenté à gauche du diagramme et le système est au milieu et l'objet PFE à droite.

- L'utilisateur commence d'abord par s'authentifier
- Si l'authentification est validée et l'utilisateur est un étudiant il devra insérer des données (codes, email, binôme), le nom de son binôme et télécharger ces données
- Sinon, si l'authentification est validée et l'utilisateur est un administrateur il recevra une liste des étudiants inscrits dans le PFE et il va leurs affecter des enseignants
- L'administrateur récupèrera la liste des informations concernant le PFE et la téléchargera

3.5 Modélisation conceptuelle et logique des données

3.5.1 Définition

Une base de données est un ensemble organisé de données. Il est la collection de tables, requêtes, procédures et autres objets. Les données sont généralement organisées pour être utilisées par des processus. Les bases de données sont souvent mises à disposition des utilisateurs des serveurs locaux pour une sécurité optimale[3].

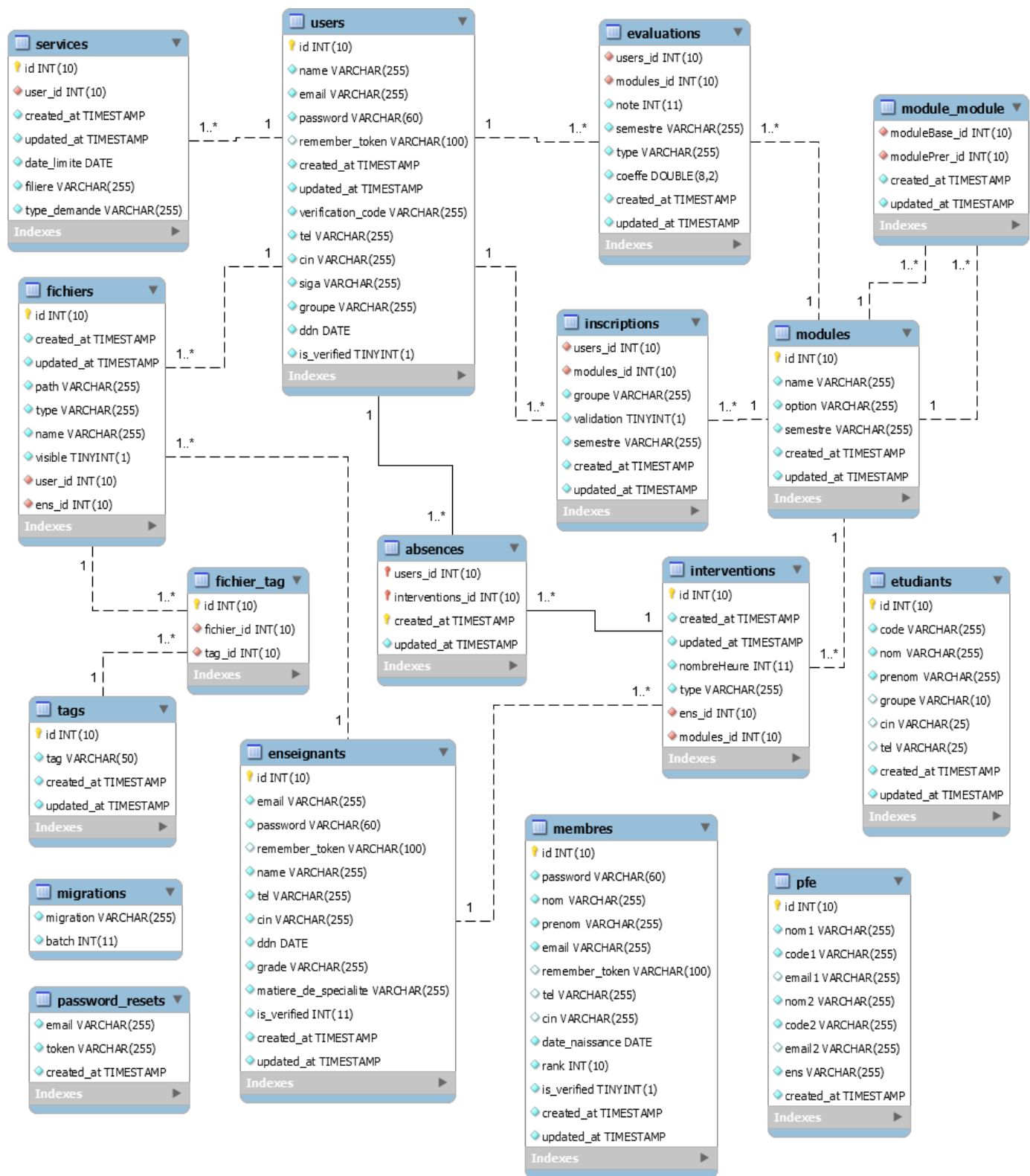
3.5.2 Modèle Conceptuel des données :

Le modèle conceptuel de données est un modèle de données pour décrire les aspects de données ou d'information dans le domaine des entreprises, de façon abstraite qui sera mis en œuvre dans une base de données comme une base de données relationnelle. Les principales composantes de ce modèle sont des entités et des relations qui peuvent exister entre eux. Il repose sur une représentation graphique qui facilite considérablement sa compréhension. La figure 3.5.1 illustre ce type de modèle

3.5.3 Modèle logique des données

Le modèle relationnel représente la base de données comme un ensemble de tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle relationnel où il est possible de relier ces structures à des tables au niveau logique. Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données en mémoire. De façon informelle, les données sont organisées sous forme de relations.

Le modèle logique correspond au diagramme de classe UML



Modèle conceptuelle de la base de données

Le schéma relationnel correspondant au modèle conceptuel précédent

- **Absences**(users->users_id, interventions_id, created_at, updated_at);
- **Enseignants**(id, email, password, remember_token, name, tel, cin, ddn, grade, matiere_de_specialite, is_verified, created_at, updated_at);
- **Etudiants**(id, code, nom, prenom, groupe, cin, tel, created_at, updated_at);
- **Evaluations**(users ->users_id, modules -> modules_id, note, semestre, type, coeff, created_at, updated_at);
- **Fichiers**(id, created_at, updated_at, path, type, name, visible, users->user_id, enseignants->ens_id);
- **Fichier_Tag** (id, fichiers->fichier_id, tags -> tag_id);
- **Inscriptions**(users -> users_id, modules -> modules_id, groupe, validation, semestre, created_at, updated_at);
- **Interventions**(id, enseignants ->ens_id, modules->modules_id, nombreHeure, type, created_at, updated_at);
- **Membres**(id, password, nom, prenom, email, remember_token, tel, cin, date_naissance, rank, is_verified, created_at,updated_at);
- **Modules**(id, name, option, semestre, created_at, updated_at);
- **Module_Module**(modules ->moduleBase_id, modules ->modulePrer_id, created_at, updated_at);
- **Password_resets**(email, token, created_at);
- **Services**(id, users->user_id, date_limite, filiere, type_demande, created_at, updated_at);
- **Tags**(id, tag, created_at, updated_at);
- **Users**(id, name, email, password, remember_token, verification_code, tel, cin, sifa, groupe, ddn, is_verified, created_at, updated_at);
- **Pfe**(id, nom1, code1, email1, nom2, code2, email2, enseignant, created_at);

3.6 Conclusion

Dans ce chapitre, nous avons présenté notre étude conceptuelle du système. La vue fonctionnelle a été illustrée par les diagrammes de cas d'utilisation. Ensuite, la vue statique, réalisée par les diagrammes de classes qui nous a permis de définir la structure du système et de dégager les différentes entités du composant, nous avons aussi proposé une solution d'une vue visuelle statique du système. Enfin, la vue dynamique nous a permis d'avoir une vue globale sur le déroulement des cas d'utilisation et leurs exécutions, cette vue a été modélisée par des diagrammes de séquence système afin de modéliser l'interaction du système avec les acteurs et décrivant ainsi son comportement. Dans le chapitre suivant, nous détaillons quelques aspects de la conception technique.

Chapitre 4

Conception technique

4.1 Introduction

Ce chapitre est décomposé en trois sections importantes. Tout d'abord, une étude préliminaire sur les technologies, Framework, architectures dédiés pour le développement web, ensuite le choix de la solution technique à adopter. Et enfin, une vue globale sur la plateforme du projet.

4.2 Environnement et technologies logicielles

4.2.1 Technologies utilisées

HTML

L'HTML est un langage informatique utilisé sur Internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie Hypertext Markup Language, ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage. L'HTML5 est le successeur de l'HTML 4.01, ça veut dire qu'il s'agit toujours du HTML à la différence de quelques nouvelles balises. De plus, la version 5 est aujourd'hui compatible avec la majorité des navigateurs et répond aux normes W3C (C'est une communauté internationale où les membres, une équipe à plein temps, et le public travaillent ensemble pour développer les standards du web) .Fondamentalement HTML5 a ses nombreuses nouvelles fonctionnalités syntaxiques, qui comprennent le `<video>`, `<audio>`, et des éléments de `<canvas>` [14]. En raison de ces nouveaux éléments, il sera très facile d'intégrer du contenu multimédia et graphique pour le Web sans utiliser le flash et plugins tiers. Il y'a aussi des nouveaux éléments comme `<section>`, `<article>`, `<header>` et `<nav>` qui enrichissent la valeur sémantique du document[14].

CSS

Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des

éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc.[14]

PHP

PHP est un langage de script coté serveur qui à été conçu spécifiquement pour le Web. Le code PHP est inclus dans une page HTML et sera exécuté à chaque fois qu'un visiteur affichera la page. Le code PHP est interprété au niveau du serveur web et génère du code HTML ou toute autre donnée affichable dans le navigateur de l'utilisateur. PHP a été conçu en 1994 par Rasmus Lerdorf. Il a ensuite été adopté par d'autres personnes talentueuses et réécrit quatre fois avant de devenir le produit abouti que nous connaissons aujourd'hui. En novembre 2007, il était installé sur plus de 21 millions de domaines et sa croissance est rapide. PHP est un projet open-source, ce qui signifie que vous pouvez vous procurer son code, l'utiliser, le modifier et le redistribuer gratuitement. La dernière version principale de PHP est la version 5. Elle bénéficie d'une réécriture complète du moteur Zend et de quelques améliorations importantes au niveau du langage[14].

JavaScript

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web[14].

JQuery

jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript[14].

MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde[14].

4.3 Framework

4.3.1 Définition

Un Framework est un outil qui regroupe tout un ensemble de fonctionnalités pré-existantes[8]. Il vous permet de bénéficier d'une architecture d'application de base et offre ainsi aux développeurs un confort de conception grâce notamment à une facilité et à une rapidité de développement, à des conventions précises et à une décomposition de l'application méthodique et logique.

4.3.2 Exemple de Framework web :

Python : Django - Flask - Tornado - Scrapy - Karrigell - Twisted - Web2py - CherryPy - Pyramid - TurboGears

PHP : Symfony Zend Framework - CodeIgniter - FuelPHP - Yii - CakePHP - QCodo - Laravel - Jelix - Agavi - Atomik - Copix - Kumbia PHP - PRADO - WebSite-PHP - Phalcon

Ruby : Ruby on Rails - Sinatra - Merb

Perl : Catalyst - Dancer - Mojolicious

JavaScript : jQuery - AngularJS - MooTools - Dojo - Meteor - Backbone.js - Ember.js - ExtJS - React - qooxdoo - YUI - Cappuccino - script.aculo.us - Prototype

.NET : Java EE - Spring MVC - Struts - Tapestry - Play - Stripes - ZK - Cocoon - GWT - Echo - AppFuse - Grails - RAP - Wicket - WebObjects - Vert.x

4.3.3 Architecture logicielle

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur) Le principe d'une telle structure est de diviser l'application en 3 parties distinctes :

- Les modèles : ils représentent les données de l'application et permettent l'interaction avec la base de données (ou les web services, etc).
- Les vues : ce sont les représentations (les templates) des résultats de la requête que l'utilisateur a effectuée.
- Les contrôleurs : ils interceptent toutes les requêtes faites par les utilisateurs.

Architecture 3-tiers : Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles (étages, niveaux, tiers ou strates) dont le rôle est clairement défini :

- la **Présentation des données** : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur.
- le **Traitement métier des données** : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- **L'accès aux données** persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

4.4 Choix technologiques

Choix de la technologie

Le langage le plus approprié pour la programmation du site, en plus de HTML qui nous servira de base, sera le PHP, lui-même combiné à un système de gestion de base de données, MySQL. Ce sont les langages que l'on utilise le plus souvent pour la programmation des sites web, car ils permettent de rendre les pages Web plus dynamiques. Tout d'abord, le PHP est un langage de programmation qui s'intègre dans les pages HTML. Il permet, entre autres, de rendre automatiques des tâches répétitives, notamment grâce à l'utilisation de fonctions qui sont en quelques sortes des petits programmes permettant d'accomplir ces tâches, et grâce à la communication avec une base de données (qui contient les informations). De plus, il fallait choisir un langage permettant de manipuler les sessions. Une session correspond à l'intervalle de temps compris entre le premier accès par un utilisateur au site web et la clôture de toutes les fenêtres de son navigateur. Les sessions permettront de stocker les informations concernant le membre connecté, PHP gère très bien les sessions et elles sont plus simples à utiliser et à mettre en œuvre.

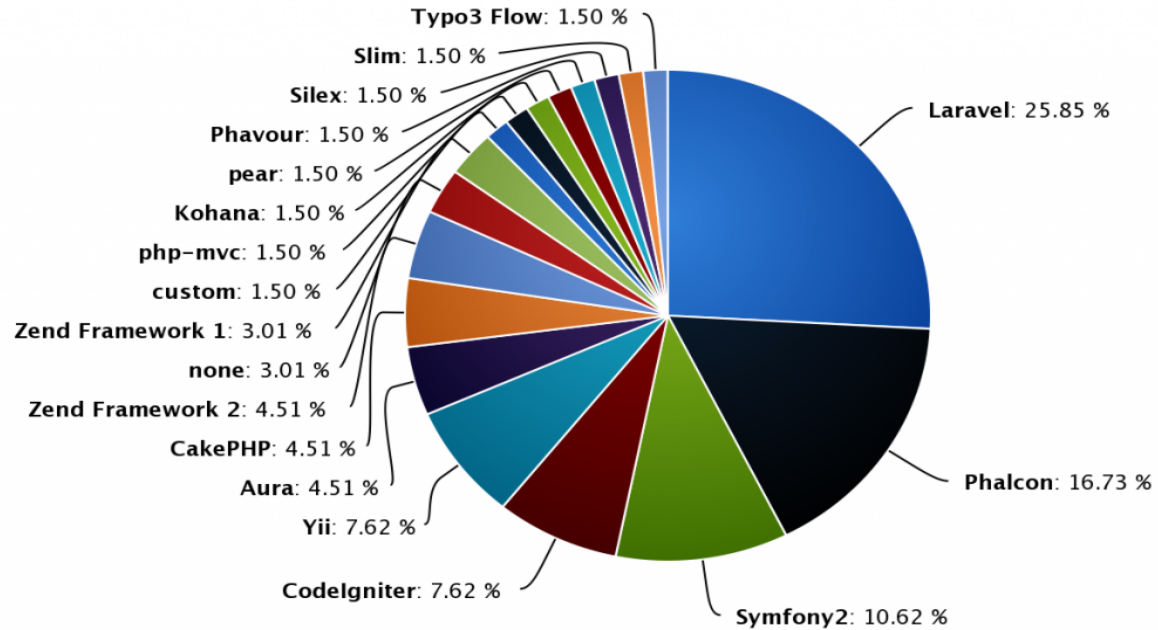
Choix du framework :

La solution retenue a été de développer l'application à l'aide d'un framework, et ceci pour la raison qu'un CMS n'est pas adapté pour le genre de base de données interactive. En effet, l'application à réaliser n'est pas un site public ouvert à tout le monde, c'est à dire que la plupart des fonctionnalités d'un CMS ne seraient pas d'avantages. Et dans l'hypothèse où l'on utiliserait un CMS, il faudrait désactiver ces fonctionnalités, et donc modifier directement le code source.

Il existe de nombreux Framework PHP open-source, c'est à dire gratuits, librement modifiables et distribuables. Voici les plus connus :

1. Zend Framework est un cadre de développement pour PHP, orienté objet. Zend étant le créateur du langage PHP, cela lui confère une grande crédibilité. Ses principales fonctionnalités sont :
 - Sécurité : système de protection contre les attaques par injections SQL et des attaques de types cross-site-scripting (XSS).
 - Séparation du code en trois couches MVC.
 - Architecture totalement modulaire, c'est à dire que l'on peut inclure uniquement les fonctionnalités dont on a besoin.Ce framework est surtout destiné aux sites à forte charge et aux développeurs web de métier qui peuvent se permettre de passer des semaines à le prendre en main.
2. Symfony est un Framework MVC open-source écrit en PHP 5, donc orienté objet. Ses principales fonctionnalités sont :
 - Une séparation du code en trois couches, selon le modèle MVC.
 - Un système de template évolué.
 - Des performances optimisées et un système de cache pour garantir des temps de réponse optimaux.

Framework popularity, end of 2013; SitePoint



Highcharts.com

Statistiques de popularité Framework PHP fin 2013

- Une couche de mapping objet-relationnel (ORM) et une couche d'abstraction de données.
- Le support de l'Ajax.

Il semble une très bonne alternative à Zend, disposant des fonctions équivalentes à celui-ci mais plus simple à maîtriser par sa conception. Cependant, il est assez lourd et lent.

3. Laravel est un framework d'application web PHP open-source, destiné pour le développement d'applications. Ses principales fonctionnalités sont :
 - *Composer* en tant que gestionnaire de dépendance pour ajouter des packages de structure différentes et spécifiques disponible à partir du référentiel Packagist permettant ainsi le support de la majorité des technologies actuelle.
 - *Eloquent ORM* (mapping objet-relationnel) est une implémentation de PHP avancé du motif d'enregistrement actif, en fournissant en même temps des méthodes internes pour appliquer des contraintes sur les relations entre les objets de base de données.
 - Laravel dispose d'un moteur de template "*Blade*" qui combine un ou plusieurs mo-

dèles avec un modèle de données pour produire des nouvelles vues, en traduisant les modèles dans le code PHP en cache pour améliorer les performances.

- *Migrations* : fournissent un système de contrôle de version pour les schémas de base de données, ce qui permet d’associer des changements dans le code source de l’application et les modifications nécessaires à la mise en page de base de données. En conséquence, le déploiement et la mise à jour des applications est considérablement simplifié.

Laravel ressemble à Symfony, disposant des fonctions équivalentes. De plus, malgré son jeune âge, il est fiable et éprouvé.

Choix du système de template

Un système de template web utilise un processeur de modèles, éventuellement à l’aide d’une source de données pour personnaliser les pages ou présenter une grande quantité de contenu sur les pages similaires prospectifs[11]. Pour notre besoin le modèle web doit satisfaire ces caractéristique :

- Une vue administrateur responsive (Adapté à toute sorte d’écran)
- de source libre et gratuite
- Construit avec Bootstrap 3 afin d’être compatible avec Laravel
- Facile à personnaliser
- Communauté très active

Nous avons adopté AdminLTE comme système de template puisqu’il est libre, gratuit, populaire et utilisé pour les tableaux de bord d’administration et de panneaux de contrôle. c’est est un template HTML sensible qui est basé sur le framework CSS Bootstrap 3. Il utilise tous les composants Bootstrap dans sa conception et modifier le style de nombreux plugins couramment utilisés pour créer une conception cohérente qui peut être utilisé comme une interface utilisateur pour les applications back-end. AdminLTE est basée sur une conception modulaire, ce qui lui permet d’être facilement personnalisé et construit. Une autre documentation vous guidera à travers l’installation du modèle et explorer les différents composants qui sont regroupés avec le modèle dans un deuxième rapport de guide d’utilisation.



Template AdminLTE

4.5 Solution Retenue

Tenant compte de cette étude, nous avons choisi de travailler avec PHP5, MySQL, Javascript, Ajax, Apache, Html5 et Css3 comme technologies puis Laravel5 et Bootstrap3 comme

frameworks ce qui nous permet d'organiser l'application avec l'architecture MVC et implémenter la solution avec l'architecture 3 tiers.

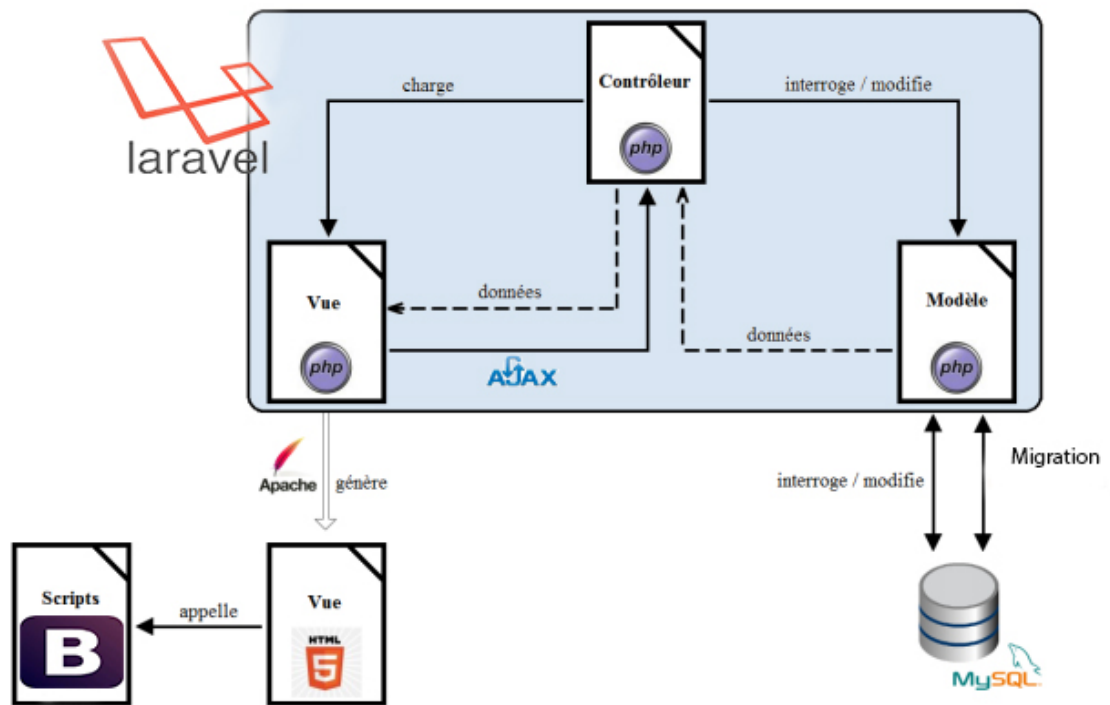


Figure Fonctionnement de l'application.

4.6 Conclusion

Nous avons présenté dans ce chapitre une idée sur le framework et l'architecture à adopter dans notre projet. Cette étude nous a permis de distinguer les différents points sur lesquels on va travailler et les technologies qu'on va utiliser pour réaliser notre application. La partie réalisation fera l'objet du chapitre suivant.

Chapitre 5

Réalisation

5.1 Introduction

Les figures ci-dessous représentent quelques captures d'écran de pages de notre site. Les pages qui suivent l'authentification contiennent tous des éléments communs d'une part le header qui contient le nom du membre sous forme d'une liste, contenant deux boutons un pour se déconnecter et fermer la session et l'autre pour pouvoir modifier des données qui lui sont personnel(nom, email, mot de passe, etc), d'autre part une sidebar, c'est-à-dire un menu latéral, qui va nous permettre de naviguer dans les différentes pages du site. Dans cette partie nous allons décrire notre réalisation sous forme d'un scénario de l'étudiant **Ilyas Amine**.

5.2 Cas : S'inscrire au système.

Chaque internaute aura la possibilité de s'inscrire dans le système, à condition qu'il soit un étudiant au sein du département informatique, ou un enseignant nous traiterons dans cette section le cas d'une inscription d'un étudiant. D'abord l'internaute devra saisir des informations personnelles, en les appuyant d'un document de type image qui identifiera l'étudiant de façon unique (exemple : acte de naissance).

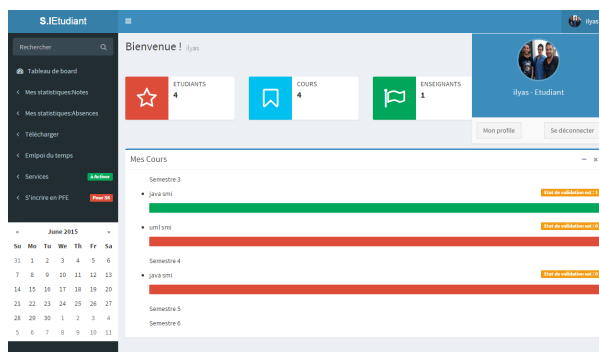


Page d'inscription dans le système. Le formulaire est intitulé "Système d'Informations des étudiants" et "Université Mohammed V Faculté des Sciences Rebat". Il contient des champs pour : Nom et Prénom, Adresse Email, Mot de Passe, Confirmer Mot de Passe, Votre Photo de profil (PSE) (Chargement ou Sélectionner un fichier), Photo image (Chargement ou Sélectionner un fichier), Téléphone, Carte d'identité nationale, Code, Groupe, et Date de Naissance. Un bouton "Valider" est présent en bas.

5.3 Cas : Suivre des cours.

Page d'inscription dans le système.

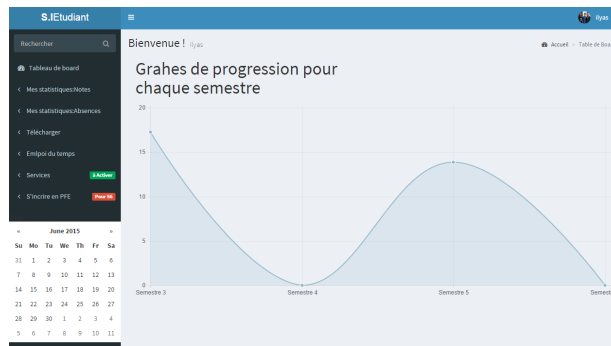
En s'authentifiant, Ilyas amine est redirigé vers la page d'accueil qui contient des informations sur le nombre d'étudiants, le nombre d'enseignants du département et les cours publiés dans le système ainsi que les cours dans lesquels l'étudiant est inscrit durant l'année universitaire courante.



Page d'accueil.

5.4 Cas : Visualiser le Graphe de notes.

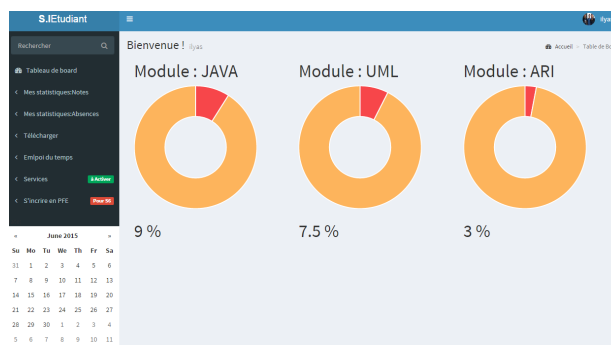
Ilyas amine peut visualiser les résultats d'un semestre en cliquant sur le liens du sidebar " Mes statistiques : Notes ", il sera redirigé, alors, vers la page des notes qui contient un graphe de progressions de la moyenne de chaque semestre.



Page des notes par semestre de l'étudiant.

5.5 Cas : Visualiser les Statistiques des absences

Ilyas amine pourra aussi voir le pourcentage d'absences dans chaque modules dans lequel il est inscrit en cliquant sur " Mes statistiques : Absences ".



Page d'absences par module de l'étudiant.

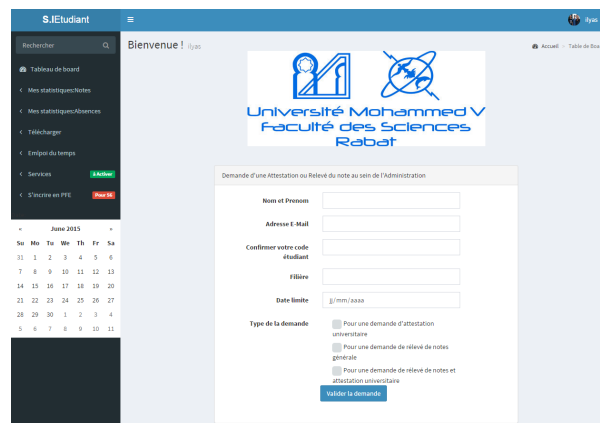
5.6 Cas : Télécharger des fichiers



Page des fichiers.

Ilyas amine pourra aussi télécharger la liste des cours soit en cliquant sur " Télécharger " et choisir le cours parmi la liste affichée soit en le recherchant par son Tag dans la barre rechercher et ensuite le télécharger.

5.7 Cas : Demander des services



Page des services.

Le système offre aussi d'autres fonctionnalités pour l'étudiant notamment la demande d'un service tel qu'une attestation de scolarité, un relevé de note, etc.

Pour la demandes d'un service, Ilyas Amine devra remplir un formulaire qui contient certaines données personnelles ainsi que le type de service. Cette demande sera traitée par la suite par un administrateur chargé du traitement des services.

5.8 Cas : S'inscrire au module de PFE

S.Etudiant

Bienvenue ! Ilyas

Accueil - Table de Bord

Votre Nom

Entrez votre nom

Entrez votre code

Entrez votre email

Entrez votre binôme

nom

code

email

Remarque

Veuillez Confirmer la procédure et imprimer la fiche pour la signer.

Valider

June 2015

Sa Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 1 2 3 4

5 6 7 8 9 10 11

Page d'inscription au module de PFE.

Concernant l'inscription à un PFE Ilyas amine devra saisir son email, son code et ceux de son binôme ensuite les imprimer. L'inscription sera traitée par un autre administrateur chargé du PFE.

5.9 Cas : Visualiser les emplois du temps.

S.Etudiant

Bienvenue ! Ilyas

Accueil - Table de Bord

Université Mohammed V
Faculté des Sciences
Rabat

Les emplois du temps : Parcours informatiques

Filière	Groupe
SIR-L2 A	A01
SIR-L2 A	A02
SIR-L2 B	B01
SIR-L2 B	B02
SIR-L3 A	A01
SIR-L3 A	A02
SIR-L3 B	B01
SIR-L3 B	B02

June 2015

Sa Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 1 2 3 4

5 6 7 8 9 10 11

Page des emplois du temps.

Ilyas Amine pourra avoir une liste des emplois du temps des différentes sections et, par conséquent, les télécharger.

5.10 Cas : Affecter des PFE.

ID	Etudiant 1	Etudiant 2	Année	Etat	Encadrant
10	Ryas	Ryas 01	2015	Approuvé	Oussama Reda
11	Ryas	Ryas 01	2015	Approuvé	Houfene Salma
12	Ryas	Ryas 01	2015	Approuvé	El Marmouh SOUJDI
13	Ryas	Ryas 01	2015	Approuvé	Said SLADOU
14	Ryas	Ryas 01	2015	Approuvé	Houad Ben Hamou
15	Ryas	Ryas 01	2015	Approuvé	H.D. RAHMANNI

Page d'affectation par l'administrateur.

Après l'authentification de l'administrateur, chargé de l'affectation du pfe, une page contenant la liste des étudiants inscrits en PFE sera affichée. Ce dernier aura pour rôle de choisir un encadrant pour chaque binôme et imprimer, par la suite, la liste finale.

5.11 Cas : Ajouter des fichiers.

Ajouter un cours

Mot clé:

Choisissez un fichier:

Page d'impression des demandes par l'administrateur.

L'administrateur des services aura pour rôle de communiquer les services demandés par les étudiants à l'administration sous format papier en imprimant la liste des demandes, il devra aussi ajouter des cours fournis par des enseignants, il pourra aussi visualiser l'identité d'un étudiant donné.

5.12 Conclusion

Nous avons vu tout au long de ce chapitre, différents scénarios de notre application, l'étape suivante sera une conclusion et perspectives dans laquelle on va résumer les différents apports de notre application

Chapitre 6

Conclusion et perspectives

Aujourd'hui, le souci majeur des facultés et établissements universitaires est la difficulté de communication entre administration, étudiant et enseignant ce qui affecte la qualité des études supérieures. La nécessité d'améliorer le service rendu pour l'étudiant s'est largement manifestée.

L'objectif visé à travers ce projet de fin d'études est de présenter un système d'information pour les étudiants du département informatique de la faculté des sciences de Rabat. Le présent manuscrit détaille toutes les étapes par lesquelles nous sommes passées pour arriver au résultat attendu. Nous avons essayé tout au long de notre travail de construire notre application incrément par incrément en utilisant la méthodologie Scrum et en se basant sur le diagramme de Gantt qui forment les bonnes pratiques d'ingénierie logicielle.

Dans un premier temps, Nous avons consacré nos réflexions à l'étude de l'existant et par comprendre suffisamment la présentation générale de notre application afin de dégager les différents besoins dont l'application est chargée d'y répondre. Ces besoins ont été bien traités et analysés dans la phase de conception fonctionnelle à l'aide de langage de modélisation UML.

L'étape suivante, était la conception technique. Durant toute cette phase, une présentation globale des technologies, frameworks et architectures utilisés par le système a été réalisée pour aboutir à une meilleure solution qui vise à aider le département à améliorer la qualité de service offerte. Ensuite nous avons présenté dans la partie réalisation quelques interfaces graphiques accompagnées de leurs descriptions.

Malgré toutes les difficultés rencontrées au niveau du processus de développement et les contraintes de temps, nous avons réussi à réaliser la totalité de notre application tout en respectant l'aspect sécuritaire et en préparant la documentation nécessaire. L'un des attraits du projet était de pouvoir travailler sur des logiciels libres à la pointe de la technologie et d'approfondir nos connaissances dans les bonnes pratiques de programmation. Cela nous a permis de nous rendre compte des différentes tâches auxquelles nous serons confronté et que parfois la connaissance seule ne suffit pas pour résoudre les problèmes. Cela nous fait dire que rien n'est acquis et que le travail d'ingénieur est une formation et un apprentissage de tous les jours.

Notre travail ne s'arrête pas à ce stade, en effet plusieurs fonctionnalités peuvent être ajoutées à notre application notamment un système pour la gestion des conversations et des

commentaires qui permettrait une communication virtuelle et instantanée, de même nous pourrions accompagner le service proposé par une application mobile qui réduirait le temps des demandes quelque soit le lieu où on se trouve.

Bibliographie

- [1] W.JASON GILMORE, *EASY LARAVEL 5* , Columbus,
- [2] BASSAM ABDOUSALAM,ELHAJHOUI SALOUA *CONCEPTION ET RÉALISATION D'UNE PLATE-FORME DE COMMERCE ÉLECTRONIQUE* , Faculté des sciences,Rabat
- [3] Modélisation en UML
<https://sites.google.com/site/moulinesalma/enseignement/cours-uml>
https://en.wikipedia.org/wiki/Unified_Modeling_Language
- [4] Application de création des maquettes
<https://moqups.com/>
- [5] Grafikart : Découverte du Framework
<http://www.grafikart.fr/formations/laravel/>
- [6] Laracasts : Laravel 5 Fondamentaux
<https://laracasts.com/series/laravel-5-fundamentals>
- [7] Documentation officielle du Framework Laravel 5
<http://laravel.com/docs/5.1>
- [8] Statistiques et informations actuelle des frameworks PHP
<http://www.sitepoint.com/best-php-frameworks-2014/>
<https://fr.wikipedia.org/wiki/Framework>
- [9] Documentation officielle du modèle AdminLTE
<https://almsaeedstudio.com/themes/AdminLTE/documentation/index.html>
- [10] Intégration du modèle AdminLTE
<https://almsaeedstudio.com/blog/integrate-adminlte-with-laravel>
- [11] Bootstrap documentation et exemples
<http://getbootstrap.com/getting-started/#examples>
[https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- [12] Méthode de développement agile Scrum
<https://en.wikipedia.org/wiki/Scrum>
- [13] Diagramme de Gantt.
https://en.wikipedia.org/wiki/Gantt_chart/
<https://www.smartsheet.com/gantt-chart-software>
- [14] Technologies Web
<https://en.wikipedia.org/wiki/PHP>

- <https://en.wikipedia.org/wiki/JQuery>
<https://en.wikipedia.org/wiki/JavaScript>
<https://en.wikipedia.org/wiki/MySQL>
<https://en.wikipedia.org/wiki/HTML5>
https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [15] Documentation et Package des graphes dessinées
<http://www.chartjs.org/docs/>
- [16] Package de génération en format PDF
<https://github.com/barryvdh/laravel-dompdf>
- [17] Solution Mailing
<http://bensmith.io/email-verification-with-laravel>
- [18] Latex exemples
https://en.wikibooks.org/wiki/LaTeX/Document_Structure
- [19] Citation et proverbe sur logiciel
<http://www.dicocitations.com/citations-mot-logiciel.php#fw3GYLLitAVhMAfJ.99>

Document créer par L^AT_EX