

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 développement d’une application web de
suivi du parcours universitaire**

Présenté par :
MOHAMED YAHYAOUI

FATIMA EZZAHRA LATAOUI

soutenu le 21 septembre 2020 devant le Jury

Mme. Ghizlane ORHANOU Professeur à la Faculté des Sciences - Rabat *Présidente*
M. Lahoucine BALLIHI Professeur à la Faculté des Sciences - Rabat *Encadrant*

Année universitaire 2019-2020

Remerciements

Nous tenons à remercier Monsieur **Lahoucine Ballihi**, pour son soutien et sa générosité considérable quant à l'offre de l'information.

Nous tenons également à adresser nos sincères remerciements à l'ensemble du corps enseignant de la FSR, pour avoir porté un vif intérêt à notre parcours, et pour avoir accordé leur temps, leur attention et leur énergie et ce dans un cadre agréable de complicité et de respect.

Enfin, nous tenons également à adresser nos plus sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin à la réalisation de ce projet.

Résumé

Ce rapport est un résultat de trois mois de travail sur ce projet de fin d'études. Notre objectif (était) est de développer une application web qui consiste à faciliter le suivi de chaque étudiant au sein de la Faculté des Sciences de Rabat, en simplifiant les tâches quotidiennes des étudiants et enseignants. En outre, la consultation des notes et des statistiques concernant les Masters et les écoles d'Ingénierie. La même chose pour l'enseignant, en plus de ces fonctionnalités, il a l'accès à la liste des modules qui l'enseigne, à celle de ces étudiants, et les notes de ces derniers quels qu'ils soient actuellement inscrits ou diplômés. Tout en garantissant pour chaque étudiant un compte personnel. Finalement, l'administrateur est celui qui est responsable de la gestion des compte et de contenu affiché dans chaque interface de l'application.

À l'égard de la conception, nous avons utilisé le langage UML. Quant à la partie technique, notre application est développée en JAVA sous le framework Spring Boot en backend, et JAVASCRIPT sous le framework Angular 9 en frontend et manipulent une base de données relationnelle MYSQL.

Mots clés : SPRING, HIBERNATE, ANGULAR, J2EE, JAVA

Abstract

This report is the result of three months of work on this graduation project. Our objective (was) is to develop a web application which consists in facilitating the follow-up of each student within the Faculty of Sciences of Rabat, by simplifying the daily tasks of the students and teachers. In addition, consultation of grades and statistics for Masters and Engineering schools. The same for the teacher, in addition to these features, he has access to the list of modules that teach him, to that of these students, and the grades of the latter whoever they are currently enrolled or graduated. . While guaranteeing for each student a personal account. Finally, the administrator is the one responsible for managing the accounts and content displayed in each interface of the application.

With regard to the design, we used UML language. As for the technical part, our application is developed in JAVA under the Spring Boot framework in backend, and JAVASCRIPT under the Angular 9 framework in frontend and handle a MYSQL relational database.

Keywords : SPRING, HIBERNATE, ANGULAR, J2EE, JAVA

Table des matières

Remerciements	i
Résumé	ii
Abstract	iii
Introduction	ix
1 Présentation et Analyse du projet	1
1.1 Introduction	1
1.2 Etude de l'existant	1
1.2.1 Contexte général du projet	1
1.2.2 Exemples des applications	1
1.3 Proposition et idée	3
1.4 Cahier de charge	3
1.4.1 Etude des besoins	3
1.5 Conclusion	5
2 Analyse et conception	6
2.1 Introduction	6
2.2 Outils et langages	6
2.3 Le choix du logiciel	7
2.4 Présentation des diagrammes	7
2.4.1 Diagramme de cas d'utilisations	7
2.4.2 Diagramme de séquences	11
2.4.3 Diagramme de classes	16
2.5 Conclusion	18
3 Architecture et environnement	19
3.1 Introduction	19
3.2 HTTP et architecture du web	19
3.3 Le modèle MVC	20
3.4 Les différentes technologies	21

3.4.1	PHP	21
3.4.2	Microsoft.Net	22
3.4.3	Java EE : J2EE	23
3.4.4	Pourquoi le choix du JAVA EE?	24
3.5	Environnement	24
3.5.1	Environnement logiciel	24
3.5.2	Environnement de développement	26
3.6	Conclusion	31
4	Réalisation et Interfaces	32
4.1	L'application web "SeQuorTalib"	32
4.1.1	Les différentes interfaces	32
4.1.2	L'affichage de la base de données sous PhpMyAdmin	40
4.2	Conclusion	42
Conclusion et perspectives		43
Bibliographie		45

Table des figures

1.1	Coursera	2
1.2	Duolingo	2
1.3	Massar	2
1.4	Idee	3
2.1	Langage de modélisation UML	6
2.2	Modélio	7
2.3	Diagramme de cas d'utilisation en premier plan	8
2.4	Diagramme use case etudiant	9
2.5	Diagramme use case enseignant	10
2.6	Diagramme use case administrateur	11
2.7	Création de compte	12
2.8	Authentification	13
2.9	Consulter les notes d'étudiant	14
2.10	Ajouter un utilisateur	15
2.11	Modifier un utilisateur	16
2.12	Diagramme de classes	17
3.1	HTTP	20
3.2	Modèle mvc	20
3.3	PHP	21
3.4	MicrosoftNet	22
3.5	J2EE	23
3.6	WampServer	24
3.7	IntelliJ	25
3.8	Webstorm	25
3.9	Postman	25
3.10	Mysql	26
3.11	Spring security	26
3.12	Spring Boot	27
3.13	Hibernate	27
3.14	HTML 5	28
3.15	CSS 3	28
3.16	Bootstrap	28

3.17 Javascript	29
3.18 Angular	29
3.19 Jhipster	31
4.1 Interface authentification	32
4.2 Création du compte	33
4.3 Message d'erreur	34
4.4 Page d'accueil administrateur	34
4.5 Liste des filières	35
4.6 Espace étudiant	36
4.7 Les comptes des étudiants	36
4.8 Les graphes de suivi	37
4.9 Notes d'un étudiant	38
4.10 Page d'accueil d'un enseignant	38
4.11 Liste des visiteurs	39
4.12 Métrique et informations machine	39
4.13 Table user	40
4.14 Table étudiant	41
4.15 Table filière	41
4.16 Table département	42

Liste des tableaux

1.1 Les rôles des acteurs	3
3.1 Comparaison entre les différentes technologie web	24

Introduction

Non seulement pour satisfaire les besoins d'une entreprise, d'une collectivité ou d'un établissement. Le développement informatique permet à ces structures de bénéficier de programmes spécialisés. Certes, ces programmes pourront répondre à leurs besoins en terme global ou plus précisément en gestion de la vie étudiantes, les notes et les emplois du temps. Tout comme rafraîchir leur rythme scolaire.

Toutefois, chaque étudiant, a fortement besoin de suivre lui-même son parcours universitaire facilement. En effet, cette application lui permettrait d'avoir une idée générale sur la carrière qu'il peut avoir après sa formation, ainsi que d'autres filières qu'ils peuvent bien lui intéresser. Le présent Rapport constitue une sorte de bilan de notre travail. On a consacré notre temps à l'élaboration des différentes étapes décrites dans ce document. Il traite quatre parties :

La première concerne la présentation du projet, la seconde présente le processus d'analyse et de conception, la troisième à pour but la discussion des différents outils et langages choisies, pour aboutir enfin à la réalisation. Le rapport est enrichi d'un ensemble de captures d'écran et de figures, pour illustrer les différentes tâches menées.

Chapitre 1

Présentation et Analyse du projet

1.1 Introduction

Ce chapitre décrit le projet afin d'avoir une meilleure compréhension de ce dernier, en exposant sa problématique ainsi que ses objectifs.

1.2 Etude de l'existant

1.2.1 Contexte général du projet

Pour réussir la période universitaire. L'étudiant a besoin d'orientation et d'accompagnement permanent tout au long des années d'enseignement. D'où, des développeurs ont décidé de créer des applications web et mobiles, aidant chaque individu dans ses études et sa vie personnelle. De sorte qu'on trouve des frameworks offrant des cours certifiés afin de renforcer la formation ainsi le cv. De même organiser les dépenses et économiser le maximum possible l'argent et sans oublier l'apprentissage de nouvelles langues. Par conséquent, l'étudiant se trouve indépendant, compétent et capable de gérer parfaitement sa vie professionnelle dans un seul coup.

1.2.2 Exemples des applications

Il est évident que dans le Web, existe une multitude d'applications offrant plusieurs services aux étudiants. Quelle que soit la gestion de vie étudiante, gestion du parcours scolaire ainsi que les formations en ligne.

Voici, quelques exemples :

Coursera

Une plate-forme d'apprentissage et d'enseignement en ligne, contenant des cours vidéos animés par des formateurs experts dans plusieurs domaines appartenant aux grandes universités du monde entier. Cette dernière offre des opportunités de suivre les cours gratuitement. Les cours sans certifications sont gratuits.



FIGURE 1.1 – Coursera

Duolingo

Duolingo est un site web et une application pour mobiles et tablettes, mais aussi en ligne via un navigateur WEB, sur ordinateur, destiné à l'apprentissage gratuit des langues. Il est conçu de telle façon que les utilisateurs progressant dans leur apprentissage contribuent à la traduction de pages web.



FIGURE 1.2 – Duolingo

Massar

Massar est un site web lancé par le ministère de l'éducation. Permet aux parents de suivre le parcours scolaire de leurs enfants. À savoir, les emplois du temps, dates d'examen, les notes ainsi que les résultats.



FIGURE 1.3 – Massar

1.3 Proposition et idée

D'après ces exemples précédents, il n'existe pas un Framework contenant les propres fonctionnalités locales d'un établissement donné ayant une vision plus claire à l'aide des statistiques. Ainsi pour répondre à ce besoin. Nous proposons une application web qui va permettre aux étudiants, comme aux enseignants d'avoir les statistiques des filières et des masters ainsi que l'accès aux grandes écoles. Permettant alors de savoir les informations d'une façon directe et pratique de leurs propres étudiants, ces derniers vont aussi avoir la possibilité d'obtenir une description bien détaillée de leurs carrières, relativement aux notes classées selon les semestres.



FIGURE 1.4 – Idee

1.4 Cahier de charge

1.4.1 Etude des besoins

Les acteurs

Acteurs	Rôle
Administrateurs	Utilisateur, responsable de gérer les comptes des étudiants et enseignants, ainsi que les filières, les modules, les établissements, les départements et les notes.
Enseignant	Utilisateur, a pour rôle de consulter le profil des étudiants et voir tous les graphes d'évolution des anciennes promos.
Etudiant	Utilisateur, a le droit de consulter son profil et voir les graphes d'évolution des anciennes promotions

TABLE 1.1 – Les rôles des acteurs

Besoins fonctionnels

Le système doit permettre à :

L'étudiant de :

- Consulter le graphe de suivi concernant sa filière.
- Consulter ses notes.
- Modifier son mot de passe.

- Modifier son mail.

L'enseignant de :

- Consulter les graphes de suivi des anciennes promos concernant ses filières.
- Voir les notes de ses étudiants.
- Modifier son mot de passe.
- Modifier son mail.

L'administrateur de :

- Ajouter/Modifier/Supprimer un enseignant.
- Ajouter/Modifier/Supprimer un étudiant.
- Ajouter/Modifier/Supprimer une note d'un étudiant.
- Ajouter/Modifier/Supprimer une filière.
- Ajouter/Modifier/Supprimer un module.
- Ajouter/Modifier/Supprimer un département.
- Ajouter/Modifier/Supprimer un établissement.

NB :

- L'accès à l'application nécessite une authentification.
- Tout acteurs doit disposer d'un login (code étudiant ou matricule enseignant) et mot de passe .

Besoins non fonctionnels

Adaptation des interfaces : Non seulement améliorer les interactions homme-machine et la facilité d'utilisation. SeQuorTalib est défini par des interfaces simples bien conçues savoir ergonomiques et intuitives pour que l'utilisateur la comprenne sans aucune complication.

La performance : Le délai de traitement(demande et réponse) est censé être logique et tolérable par l'utilisateur vu que c'est une notion qui comporte des aspects techniques, mais aussi des aspects liés à l'ergonomie de l'interface entre l'utilisateur et le système. L'application doit garantir l'intégrité et la confidentialité des données.

La capacité : L'application doit être capable de stocker et gérer un grand nombre de données.

La sécurité et fiabilité : La fiabilité de notre application passe nécessairement par la prise en compte des éléments de sécurité indispensables : la sécurité autour des matériels et des logiciels, La sécurité autour des droits d'accès et habilitations, la sécurité autour des données et de leur sauvegarde.

1.5 Conclusion

Cette étape a permis de mieux définir le périmètre du projet puisqu'on a décrit la problématique puis on a établi le cahier des charges. Le chapitre suivant est consacré à l'analyse des besoins du projet.

Chapitre 2

Analyse et conception

2.1 Introduction

La conception demeure importante et indispensable dans le cycle de vie d'une application. Elle consiste à donner une image claire du déroulement d'un Framework. Cette phase a pour but donc de définir les objectifs d'après sa réalisation et assurer une bonne compréhension des besoins des utilisateurs.

2.2 Outils et langages

Afin de bien présenter l'architecture de notre application, nous avons choisi le langage de modélisation **UML** (Unified Modeling Language). Il fournit des diagrammes pour représenter le logiciel développer son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées.



FIGURE 2.1 – Langage de modélisation UML

2.3 Le choix du logiciel

L'outil adopté dans la réalisation de la conception est le logiciel Modelio, vu sa richesse fonctionnelle et ergonomique, et qui se manifeste dans :

- Il intègre également la modélisation BPMN(Model and Notation).
- Il intègre le support de la modélisation des exigences, du dictionnaire, des règles métier et des objectifs.
- Propose une gamme d'outils étendant ses fonctionnalités permettant, entre autres, la mise en œuvre de l'approche MDA(L'architecture dirigée par les modèles).

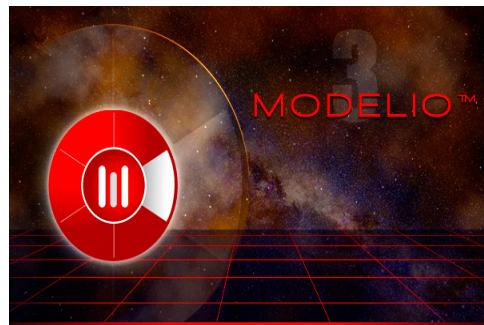


FIGURE 2.2 – Modélio

2.4 Présentation des diagrammes

2.4.1 Diagramme de cas d'utilisations

Ce diagramme vient pour répondre à la question suivante : "A quoi va servir cette application ?"

Souvent les utilisateurs ne sont pas forcément des informaticiens. Il leur faut donc un moyen simple afin d'exprimer leurs besoins et leurs objectifs. Et c'est ça exactement le rôle des diagrammes de cas d'utilisation. Car ils permettent de recueillir, analyser et organiser les besoins, et de montrer les grandes fonctionnalités du système d'information à réaliser.

D'après le cahier des charges qu'on a présenté dans le premier chapitre, voici les diagrammes de cas d'utilisations qu'on a pu dégager :

Diagramme use case en premier plan

Etudiant, Enseignant et Administrateur chacun de ces acteurs a son accès spécifique à l'application tout en exigeant une authentification .

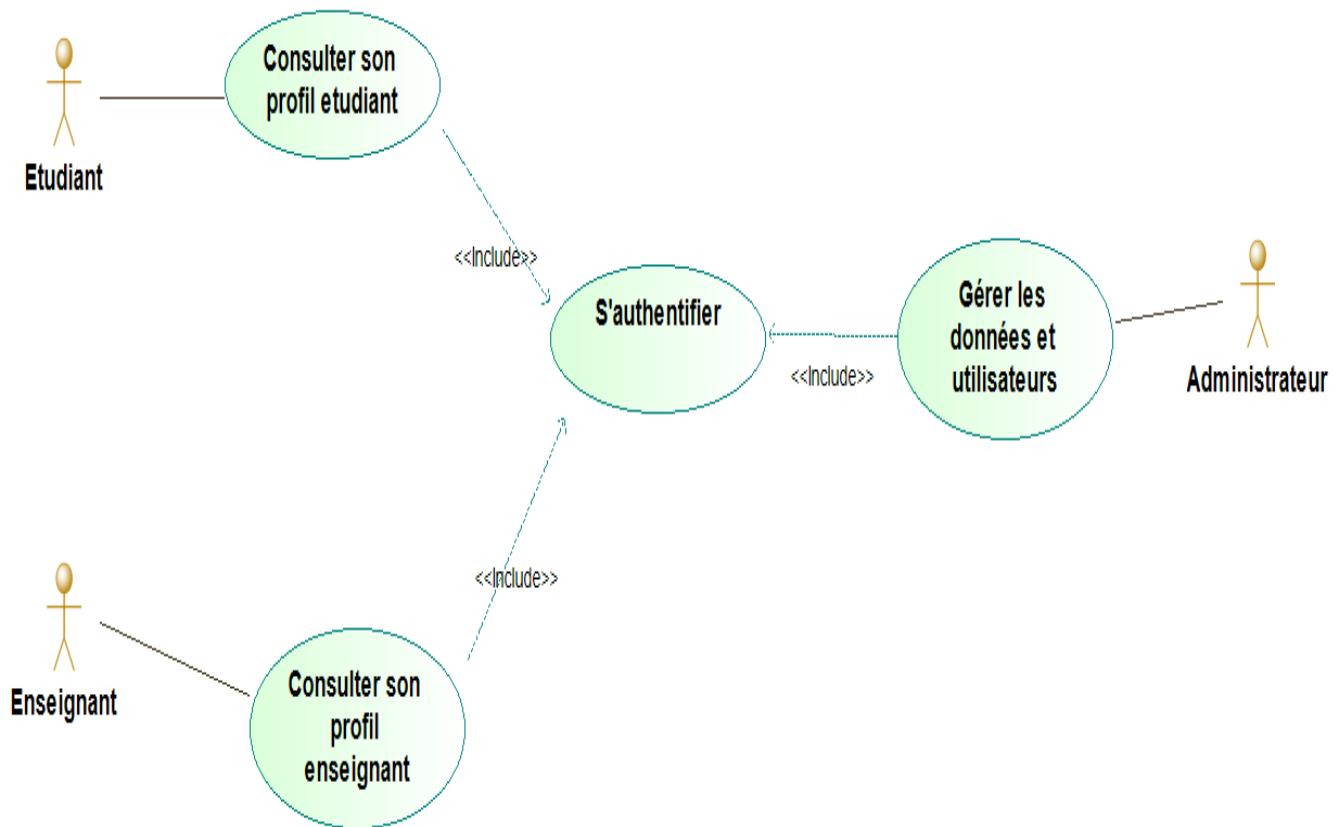


FIGURE 2.3 – Diagramme de cas d'utilisation en premier plan

Maintenant, on va découvrir les diagrammes de use case de chaque acteur.

Diagramme de use case : Etudiant

Un Etudiant, consulte ses notes et les graphes associés à sa filière.Aussi, il peut modifier son mot de passe et son email, tout en s'authentifiant.

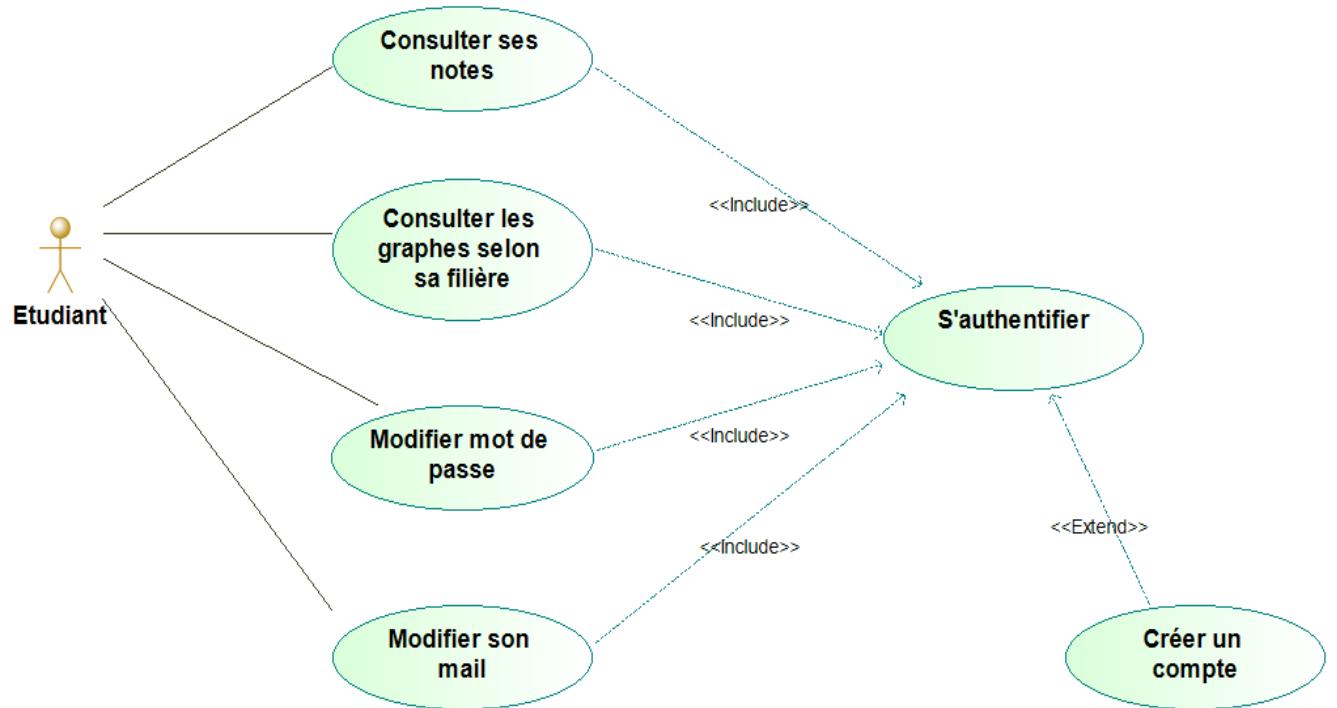


FIGURE 2.4 – Diagramme use case etudiant

Diagramme de use case : Enseignant

Autant que l'étudiant, l'enseignant à un accès à la liste de ses étudiant en consultant leurs profils ainsi que leurs notes, au moment où il s'authentifie.

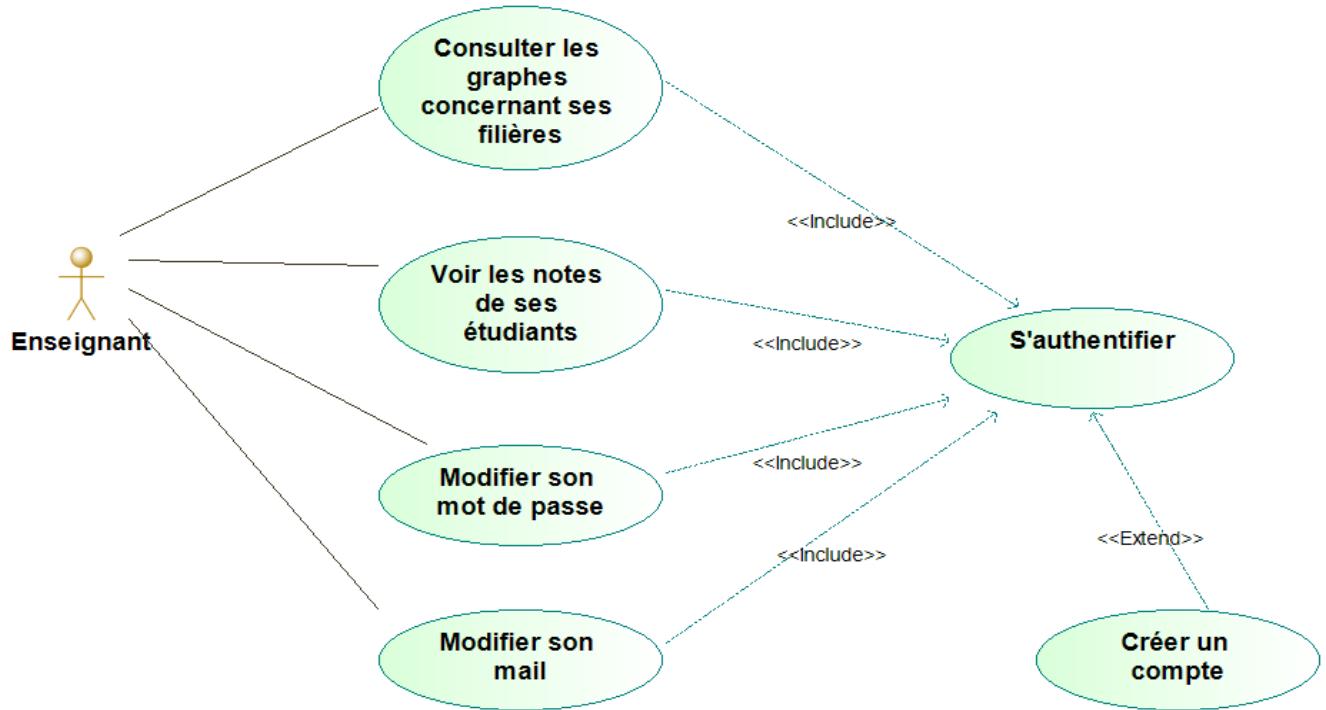


FIGURE 2.5 – Diagramme use case enseignant

Diagramme de use case : Administrateur

L'administrateur a deux rôles principaux : Premièrement, la gestion des filières, modules, notes, départements et établissements. A savoir, l'ajout, la modification et la suppression. Deuxièmement, la gestion des utilisateurs, c'est à dire ajouter, modifier et supprimer les comptes. Sans oublier la nécessité de l'authentification de cet acteur.

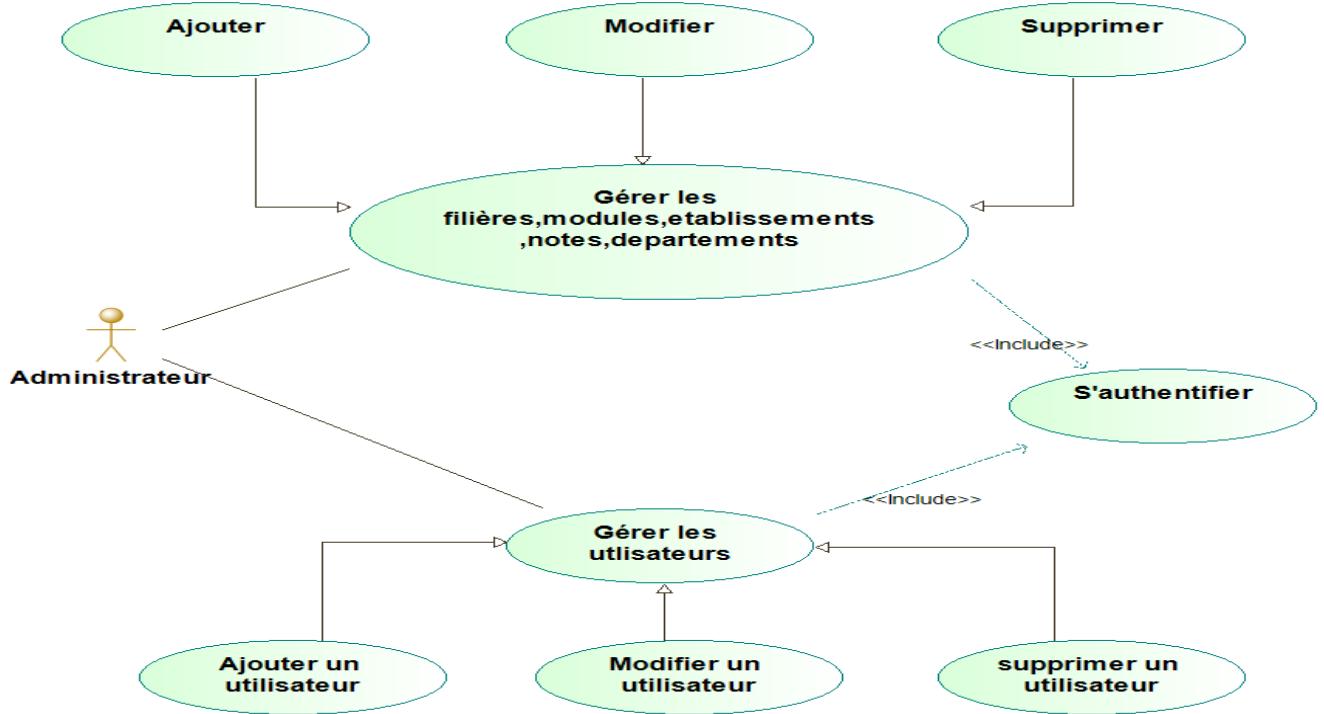


FIGURE 2.6 – Diagramme use case administrateur

2.4.2 Diagramme de séquences

Les diagrammes de séquences viennent pour répondre à la question suivante : "Comment les éléments du système interagissent entre eux et avec les acteurs ?"

A moins que le système à modéliser soit extrêmement simple, nous ne pouvons pas modéliser la dynamique globale du système dans un seul diagramme. Nous ferons donc appel à un ensemble de diagrammes de séquences chacun correspondant à une sous fonction du système, généralement d'ailleurs pour illustrer un cas d'utilisation.

Ainsi, on va présenter à titre indicatif quelques exemples de diagrammes de séquence décrivant quelques scénarios d'utilisation :

Cas d'utilisation : Crédation de compte

- Acteur : Enseignant, Etudiant.
- Aperçu : Dans le but d'accéder à l'application, l'utilisateur doit avoir un compte.
- Pré condition : L'utilisateur doit saisir son login(code), mail et password.

- Cas nominal : L'utilisateur saisi ses informations. Une requête est envoyée pour voir si l'acteur existe dans la base de données de la faculté. Si oui, l'administrateur doit activer son compte dans un délai de 48h.
- Cas d'erreur : Informations erronés, utilisateur inexistant. Rien n'est envoyé.

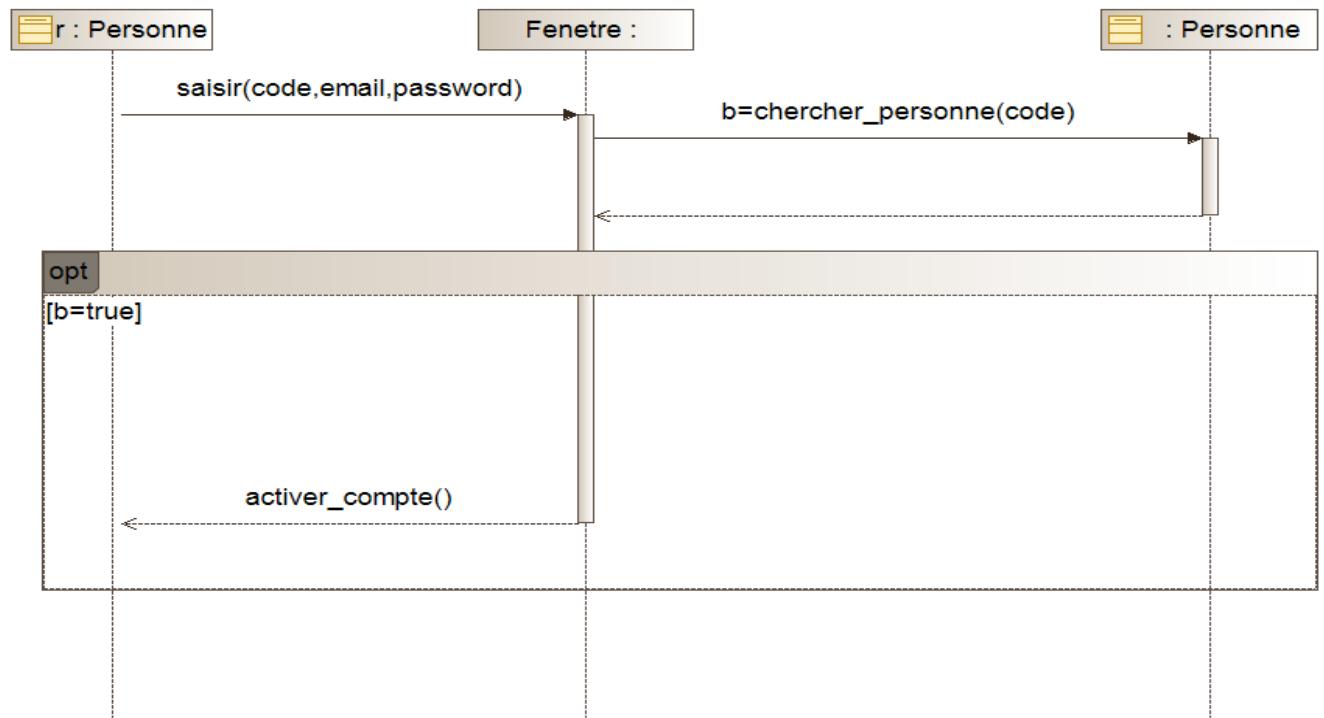


FIGURE 2.7 – Crédit de compte

Cas d'utilisation : Authentification

- Acteur : Administrateur, Enseignant, Etudiant.
- Aperçu : Dans le but d'accéder à l'application, l'utilisateur doit s'authentifier.
- Pré condition : L'utilisateur doit saisir son login et password.
- Cas nominal : L'utilisateur saisi ses informations d'identification. Une requête est envoyée pour vérifier la validité des données. Si l'authentification réussie, le système l'oriente vers sa page profil, sinon, un cas d'erreur est levé.
- Cas d'erreur : Informations erronées, utilisateur inexistant. Un message d'erreur est affiché, le système oriente l'acteur vers la page d'authentification.

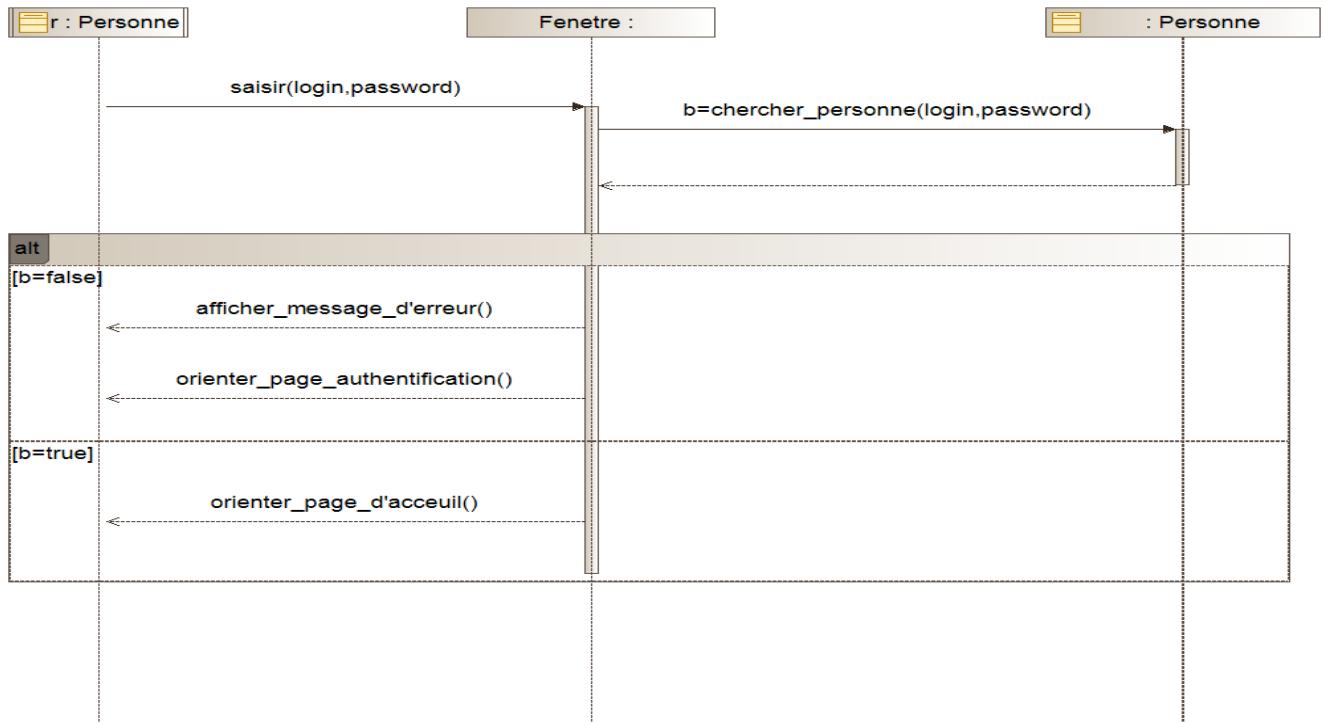


FIGURE 2.8 – Authentification

Cas d'utilisation : Consulter les notes d'un étudiant :

- Acteur : Enseignant.
- Aperçu : Dans le but d'avoir une vision générale sur ses étudiant, l'enseignant peut consulter leur profil ainsi leurs notes.
- Pré condition : L'enseignant doit être connecté.
- Cas nominal : L'enseignant saisi le code de l'étudiant. Une requête est envoyée pour faire la recherche. Si la recherche réussie, le système l'oriente vers la page profil demandé, sinon, un cas d'erreur est levé.
- Cas d'erreur : Code erroné, étudiant inexistant. Un message d'erreur est affiché, le système oriente l'acteur vers la page de saisie.

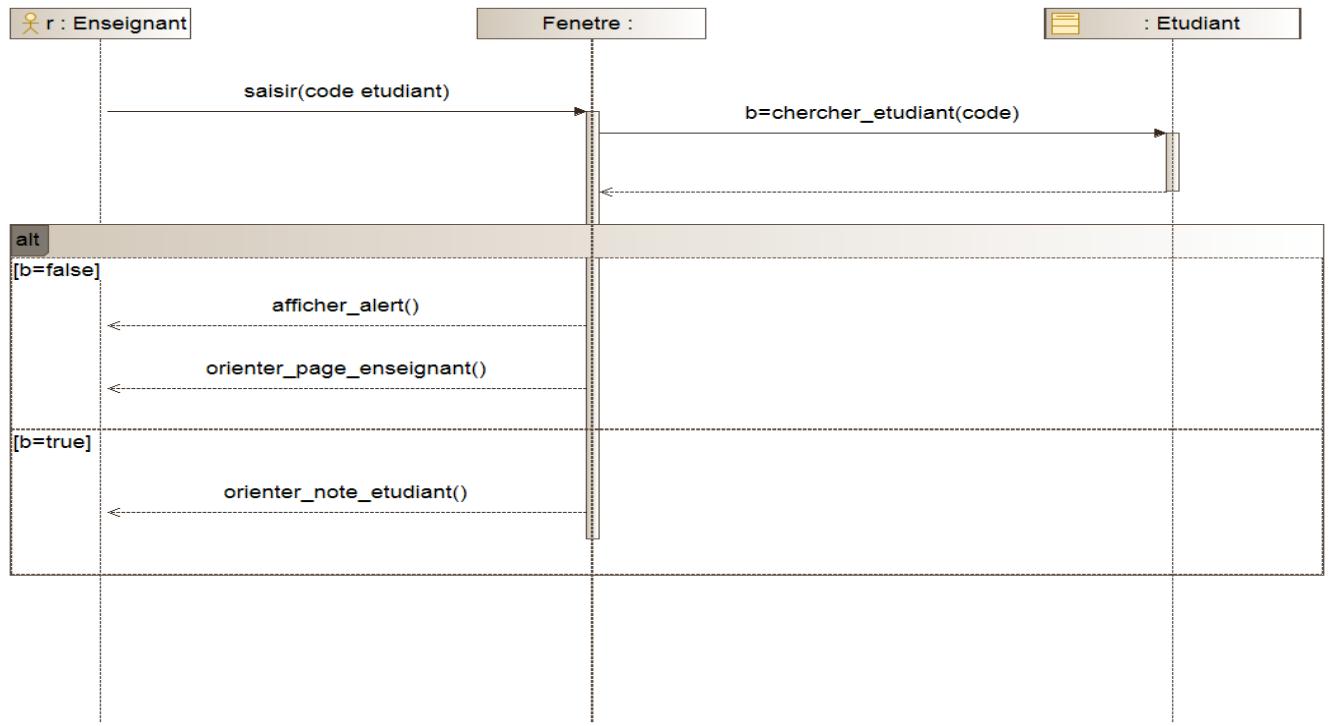


FIGURE 2.9 – Consulter les notes d'étudiant

Ajouter un utilisateur :

- Acteur : Administrateur.
- Aperçu : Afin d'assurer une bonne gestion du système, l'administrateur doit pouvoir ajouter les utilisateurs.
- Pré condition : L'administrateur doit être connecté.
- Cas nominal : L'administrateur saisi toutes les informations concernant un utilisateur. Une requête est envoyée pour effectuer l'ajout. Si l'insertion n'a pas réussi, un cas d'erreur est levé.
- Cas d'erreur : erreur système. Un message d'erreur est affiché, le système oriente l'acteur vers la page d'ajout.

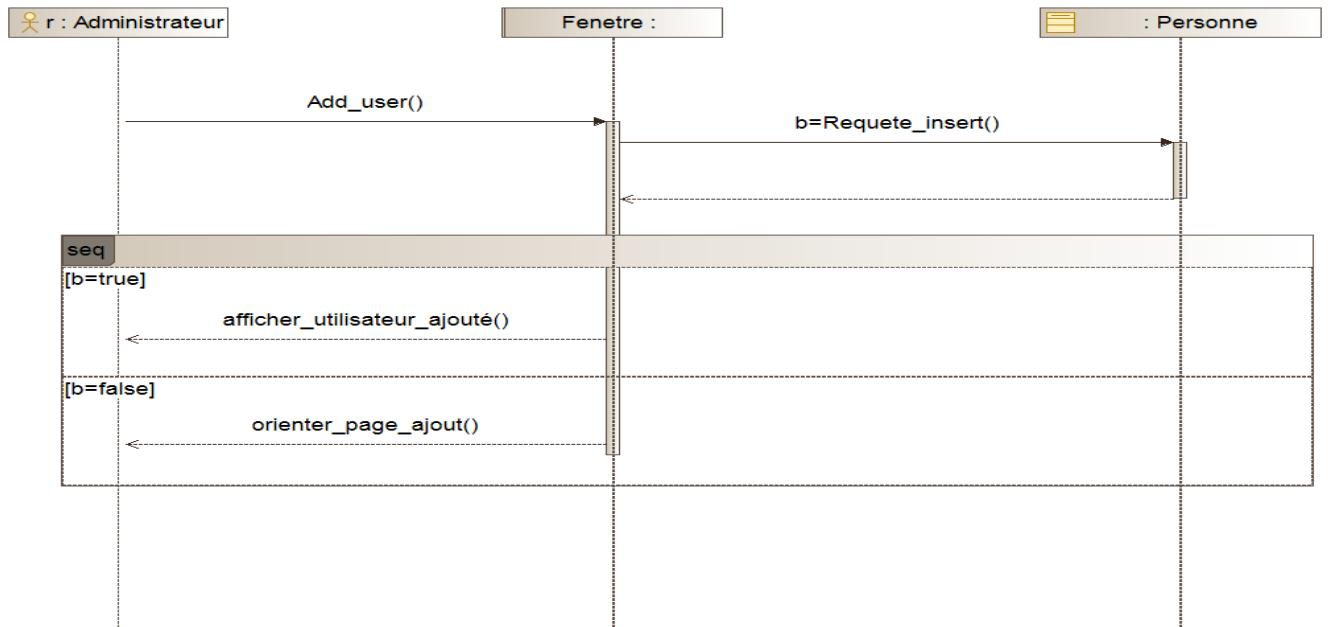


FIGURE 2.10 – Ajouter un utilisateur

Modifier un utilisateur :

- Acteur : Administrateur.
- Aperçu : Afin d'assurer une bonne gestion du système, l'administrateur doit pouvoir modifier les données des utilisateurs.
- Pré condition : L'administrateur doit être connecté.
- Cas nominal : L'administrateur consulte la liste des utilisateurs et choisit celui qui fera l'objet d'Update. Il effectue tout changement nécessaire. Une requête est envoyée pour exécuter la modification. Si la modification n'a pas réussi, un cas d'erreur est levé.
- Cas d'erreur : erreur système. Un message d'erreur est affiché, le système oriente l'acteur vers la page de modification.

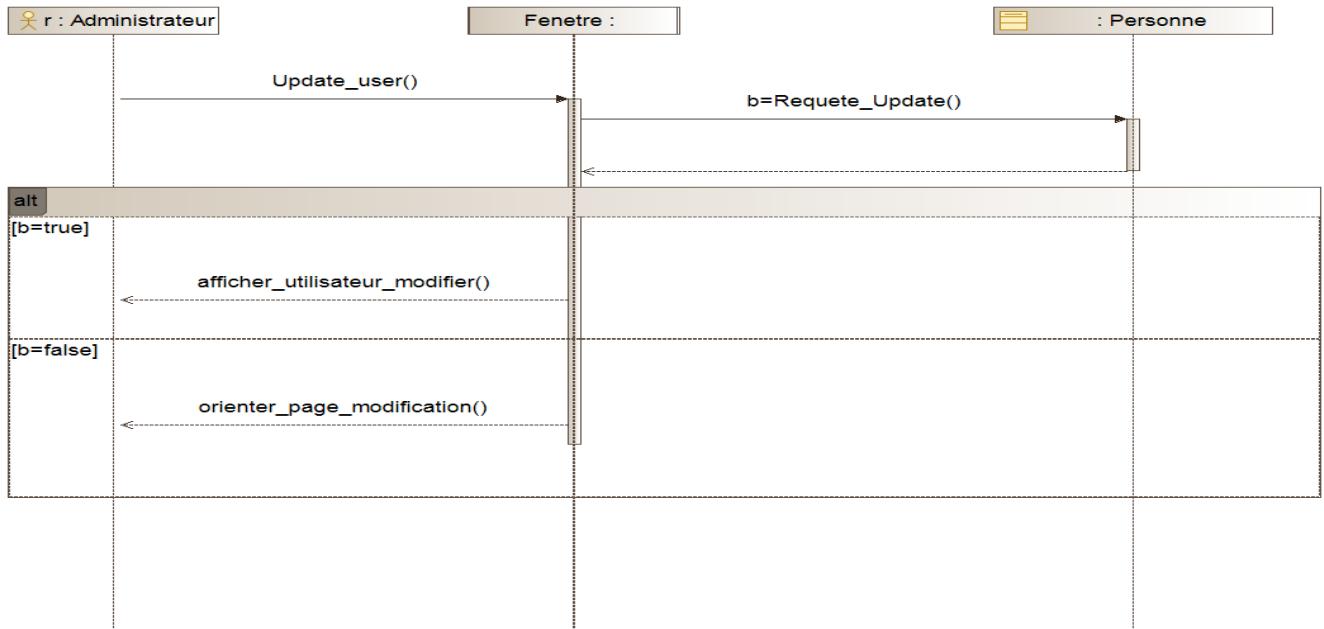


FIGURE 2.11 – Modifier un utilisateur

2.4.3 Diagramme de classes

Ce diagramme vient pour répondre à la question suivante : "Qui intervient à l'intérieur du système ?"

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système.

La description et le schéma dans la page suivante représente notre diagramme de classes :

- La classe `personne` résume trois rôles, enseignant, étudiant, et administrateur.
- L'étudiant est inscrit dans plusieurs modules, et dans une seule et unique filière.
- L'enseignant peut enseigner un ou plusieurs modules dans une ou plusieurs filières.
- Une filière possède plusieurs modules.

- Un administrateur gère plusieurs enseignants et des étudiants.
- Pour un étudiant diplômé, on doit savoir son établissement actuel ainsi que sa filière.
- Dans un département, il y a plusieurs filière et plusieurs enseignants.

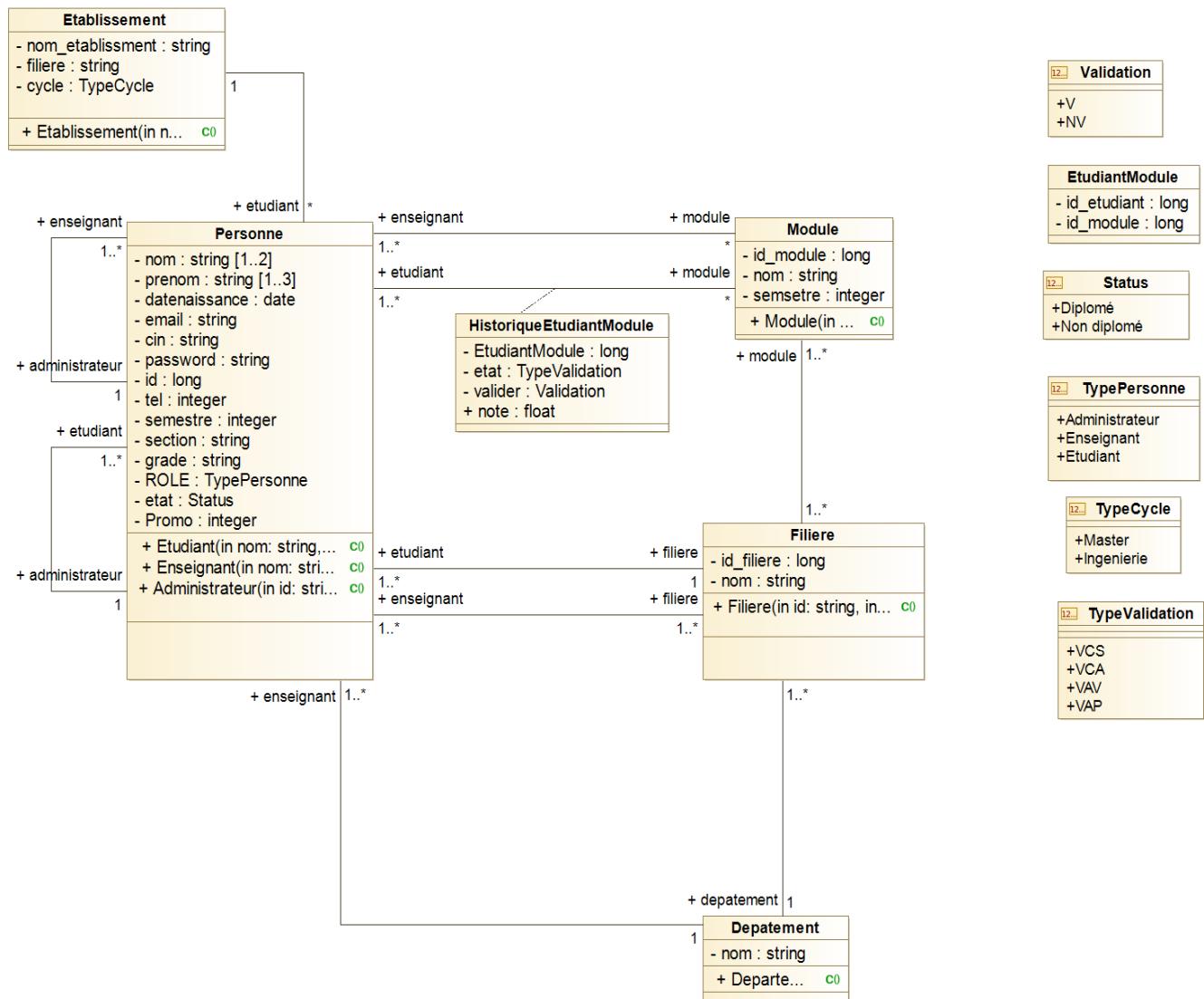


FIGURE 2.12 – Diagramme de classes

2.5 Conclusion

Ce chapitre résume le travail effectué lors des étapes d'analyse et de conception de la solution mise en place. D'abord, on a présenté les acteurs du système en décrivant leurs rôles et leurs principales tâches. Ensuite, on a décrit les différentes fonctionnalités du système à travers les différents diagrammes d'UML pour mieux comprendre les fonctionnalités offertes par le projet.

Le chapitre suivant décrira l'architecture de l'application et l'environnement de développement.

Chapitre 3

Architecture et environnement

3.1 Introduction

Il est évident que pour réussir le développement d'une application, il est nécessaire de choisir les bons outils et technologies. De même les environnements qui garantissent une excellente implémentation.

3.2 HTTP et architecture du web

HTTP : HyperText Transfer Protocol est un protocole de communication client-serveur développé pour le World Wide Web (www). HTTP a été inventé par Tim Berners-Lee avec les adresses web (URL) et le langage HTML pour créer le World Wide Web. Il utilise généralement le port 80 et est un protocole non orienté connexion. Schématiquement, le client envoie une requête au serveur qui lui retourne une réponse généralement sous la forme d'un fichier HTML. Le client affiche alors la réponse. Cette réponse contient généralement des liens hypertextes ou des formulaires, qui une fois cliqués ou soumis génèrent une requête au serveur.

- Serveur http : par exemple, Apache, IIS, sont des serveurs installés sur des machines.
- Client http : appelé aussi client léger ou navigateur Web, les plus connus sont Firefox, Internet Explorer, Opera.

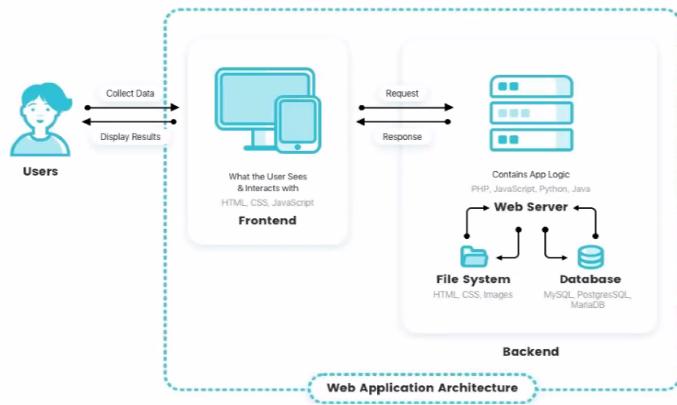


FIGURE 3.1 – HTTP

3.3 Le modèle MVC

Le **MVC** «modèle vue contrôleur» est une architecture de développement visant à séparer le code source en modules. En effet, ce modèle très répandu, consiste à séparer distinctement l'accès aux données (bases de données), la vue affichée à l'utilisateur et la logique métier. Cette architecture est le plus communément retrouvée au sein d'applications web mais existe également au niveau des applications lourdes.

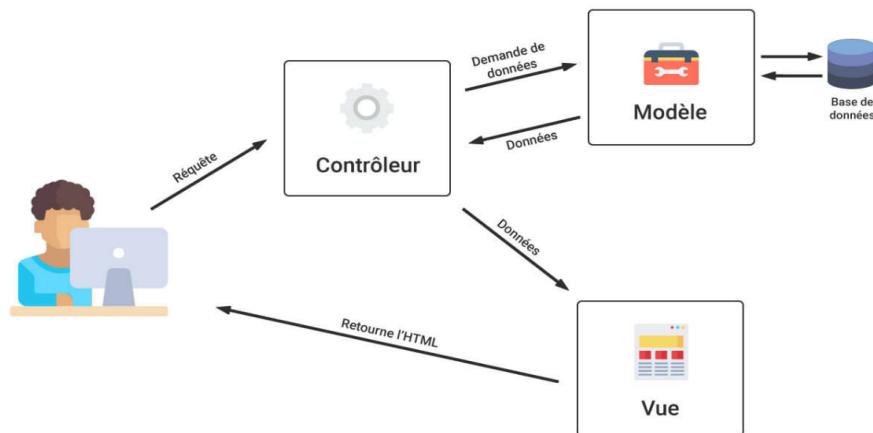


FIGURE 3.2 – Modèle mvc

3.4 Les différentes technologies

Le développement d'une application web repose sur un modèle de communication « client/serveur ». Cela nécessite l'utilisation de certaines technologies web côté client et d'autres différentes côté serveur.

3.4.1 PHP

Le PHP est un langage de programmation coté serveur. Il s'exécute entièrement sur le serveur qui héberge le site Web. Bien que son exécution est moins rapide en raison du temps que prend la requête pour parvenir au serveur, ainsi que la réponse pour arriver jusqu'au navigateur. Certes, PHP dispose de plusieurs atouts que nous allons résumer dans les points suivants :

- **Code source confidentiel** : Le code source n'est jamais visible par le client. Ce qui permet de manipuler des données confidentielles.
- **Multi-plateform** : PHP s'exécute sur des serveurs d'applications que l'on peut installer sur de nombreux systèmes d'exploitation (Unix/Linux, Windows, Mac OS).
- **Interfaçage avec les bases de données** : PHP peut interagir avec presque tous les SGDB connus.



FIGURE 3.3 – PHP

3.4.2 Microsoft.Net

Le .NET est créée, pour le développement d'applications d'entreprises multi-niveaux, basées sur des composants. D'abord, la plate-forme .NET a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ensuite, ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plates-formes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation.

L'environnement d'exécution :

- Client http : Appelé navigateur Web, Firefox, Internet Explorer, Opera sont les plus connus.
- Un moteur d'exécution, appelé **CLR** <Common Language Runtime>. Son rôle est de compiler.
- Un environnement d'exécution d'applications et de services web, appelé **ASP .NET**.
- Un environnement d'exécution d'applications lourdes, appelé **WinForms**.
- Des services, sous forme d'un ensemble hiérarchisé de classes appelé **Framework Class Library FCL**).



FIGURE 3.4 – MicrosoftNet

3.4.3 Java EE : J2EE

Java EE (J2EE) est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- Un ensemble de spécifications. à savoir qu'un tel environnement se nomme serveur d'applications.
- Un ensemble d'API qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

L'utilisation de J2EE pour développer et exécuter une application offre plusieurs avantages :

- Une architecture d'applications basée sur les composants. Donc, une séparation des rôles lors du développement
- La possibilité d'interagir avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA

Alors, J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. C'est-à-dire, ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE. Cependant, L'architecture d'une application se découpe idéalement en au moins trois tiers : La partie client, la partie métier et la partie données.



FIGURE 3.5 – J2EE

3.4.4 Pourquoi le choix du JAVA EE ?

Pour répondre à cette question, voici une simple comparaison entre les technologies web : JAVA EE – .NET-PHP.

	Java JEE	.NET	PHP
Orienté objet	Oui	Oui	Parfois
Niveau d'abstraction	Haut	Haut	Moyenne
Portabilité	Oui	Oui	Oui
Libre	Oui	Non	Oui
Puissance	Plus puissant	Plus puissant	Moyenne
Dynamisme	Fixe	Fixe	Dynamique
Hébergement	Trop chère	Trop chère	Moins chère
IDE de développement	Gratuit/Payant	Payant	Gratuit

TABLE 3.1 – Comparaison entre les différentes technologie web

D'après le tableau précédent JAVA EE et .net sont des langages de haut niveau et plus adaptés aux applications d'entreprise. En effet, elles sont très riches et plus puissantes. En addition, JAVA EE a des éditeurs d'environnement gratuits. Par contre le .net ses éditeurs sont tous payants. Par conséquent, le choix du J2EE au lieu de .NET s'impose.

3.5 Environnement

3.5.1 Environnement logiciel

WampServer

WampServer est tout d'abord une plateforme de développement Web de type WAMP, qui sert à faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Ensuite, elle dispose d'une interface d'administration sert à gérer et d'administrer ses serveurs à travers d'un *tray icon* (icône près de l'horloge de Windows).



FIGURE 3.6 – WampServer

IntelliJ

IntelliJ est un environnement de développement intégré (Integrated Development Environment - IDE) de technologie destiné au développement de logiciels informatiques. Il est développé par JetBrains, comme il es disponible en deux versions. La première, est une communautaire (open source) sous licence Apache 2 et la deuxième est une propriétaire, protégée par une licence commerciale.



FIGURE 3.7 – IntelliJ

Webstrom

WebStorm est un environnement de développement<IDE> pour les langages Web : HTML, CSS et Javascript Il est également développé par JetBrains et basé sur la plateforme IntelliJ IDEA.



FIGURE 3.8 – Webstorm

Postman

Postman est un logiciel qui se focalise sur les tests des API. Il est devenu très populaire pour tester les Microservices, notamment grâce à sa simplicité et ses fonctionnalités très spécialisées.



FIGURE 3.9 – Postman

3.5.2 Environnement de développement

Mysql

MYSQL est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde. Sa première version est apparue en 1995 et il est régulièrement entretenu. De plus, ce système est particulièrement utilisé par les développeurs pour faire en implémentant des paquetages tout prêts pour Windows(WAMP) , Mac(MAMP) et Linux(XAMPP) . Ces packages sont si populaires et simples à mettre en oeuvre que MySQL est largement connu et exploité comme SGBD, pour des applications utilisant PHP. C'est d'ailleurs, pour cette raison que la plupart des hébergeurs web proposent PHP et MySQL. La réPLICATION est possible avec ce système et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes des données.



FIGURE 3.10 – Mysql

Spring Security

Spring Security est un framework de sécurité léger qui fournit une authentification et un support d'autorisation afin de sécuriser les applications Spring. Il est livré avec des implémentations d'algorithmes de sécurité populaires.



FIGURE 3.11 – Spring security

Spring Boot

Spring est un formidable outil quand on essaye de développer une application web. Dit projet ou micro framework qui facilite le développement d'applications

fondées sur Spring en offrant des outils permettant d'obtenir une application pa-cagée en jar , totalement autonome. Il a notamment pour but de faciliter la configura-tion d'un projet Spring, et de réduire le temps alloué à son démarrage.



FIGURE 3.12 – Spring Boot

Hibernate

Hibernate, est une solution open source de type ORM <Object Relational Mapping> qui permet de faciliter le développement de la couche persistance d'une application. Permet aussi de représenter une base de données en objets Java et vice versa. De plus, Hibernate facilite la persistance et la recherche de données dans une base de données, en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. Alors, La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement redondant.



FIGURE 3.13 – Hibernate

HTML

Le **HTML5** - 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. Lancée en octobre 2014, cette version apporte de nouveaux éléments et de nouveaux attributs par rapport à la version précédente. Elle offre par exemple la possibilité de définir le contenu principal d'une page Web, d'ajouter une introduction **en header** et d'insérer un sous-titre à un contenu multimédia de type vidéo. Ainsi que d'autres fonctionnalités.



FIGURE 3.14 – HTML 5

CSS

Le **CSS 3** -Cascading Style Sheets-, servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence comme : couleurs, bordures et polices. Ainsi que de placement tel : largeur, hauteur et côté à côté. Le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.



FIGURE 3.15 – CSS 3

Bootstrap

D'emblée, Bootstrap est un framework CSS et Javascript permet de créer rapidement des éléments de pages web, avec une mise en forme par défaut inspirée du design de Twitter. En addition, La création des éléments (Responsive Design) représente l'avantage le plus important de ce framework. Enfin, le terme (Responsive Design) signifie que les éléments d'une page web créée en utilisant le framework Bootstrap, s'adaptent aux différentes tailles d'écrans. Ce qui permet un affichage sur de nombreux appareils. A titre d'exemple, les projecteurs, les écrans de bureau, les ordinateurs portables et les tablettes.



FIGURE 3.16 – Bootstrap

Javascript

Il s'agit d'un langage orienté objet exécuté dans le navigateur et qui permet de créer des animations sur les éléments présents dans une page web en fonction des actions de l'utilisateur. De plus, cela est devenu presque une obligation car il permet à l'utilisateur de mieux comprendre ce qu'il doit faire lorsqu'il s'agit d'une page contenant plusieurs conditions d'utilisation, par exemple des conditions de saisie dans certains champs d'un formulaire, rendre un bouton incliquable tant que l'utilisateur n'a pas saisi une information nécessaire.



FIGURE 3.17 – Javascript

Angular

Développé par Google, Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web. Et plus particulièrement ce qu'on appelle des « Single Page Applications ». C'est-à-dire, des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

De plus, ce Framework est basé sur une architecture de type : MVC. Permet donc de séparer les données, le visuel et les actions pour une meilleure gestion et assure une forte maintenabilité ainsi qu'une amélioration du travail collaboratif.



FIGURE 3.18 – Angular

Générateur d'application : JHipster

JHipster rassemble dans la plateforme du même nom certains des frameworks les plus appréciés pour le développement des parties serveur et client. Cette combinaison de Spring Boot, AngularJS, Bootstrap et de nombreuses autres composantes

est disponible entièrement en open source et convient parfaitement pour la création des applications Web modernes et de microservices. Ainsi qu'il fournit des services standards d'une application web. A titre d'exemple : L'authentification, création d'un compte et la traduction. Alors pour utiliser JHipster, il est indispensable d'installer : Nodejs, Yeoman, Maven ou Gradle sans oublier un JDK. Cependant, il existe également une image Docker pour tester facilement. Nous citons quelques avantages du JHipster : Tout d'abord, la collection JHipster se démarque notamment par le fait que les différentes composantes s'accordent parfaitement les unes avec les autres. L'utilisation de cette suite doit être prise en considération si on souhaite programmer son projet Web avec Java. Ainsi qu'il réduit les obstacles de départ liés au développement, simplifie et uniformise l'intégralité du processus de programmation, que ce soit pour le développement de l'application ou pour les révisions et extensions ultérieures du code. De plus, l'utilisation de cette suite de frameworks fait gagner un temps considérable. Enfin, JHipster permet également de générer en un temps record une application web, et dispose d'un code source clairement structuré. Et d'autres fonctionnalités telles que la gestion des utilisateurs ou l'internationalisation sont également disponibles.

JHipster a créé le cadre de base d'une application Web et enregistré le code correspondant dans le répertoire créé au préalable. À partir de celui-ci, on peut démarrer sa propre instance de serveur basée sur Spring Boot. La commande à utiliser pour y parvenir dépend de l'outil choisi pour la conception du serveur. Si le choix s'est porté sur Maven, on exécute les composantes du serveur en entrant la commande suivante : ./mvnw (sous macOS/Linux) ou mvnw(sous Windows).

Cependant, pour la connexion au compte administrateur, la barre de menu de l'instance de serveur est complétée avec les deux points « Administration » et « Entités ». Le premier servira à gérer l'application web. On dispose alors des options suivantes, nous citons quelques unes :

Gestion des utilisateurs : on peut créer, éditer et supprimer les profils des utilisateurs. Il est possible de répartir des rôles de cette façon. Audits : les audits donnent un aperçu des tentatives de connexion réussies (AUTHENTICATION SUCCESS) et abortées (AUTHENTICATION FAILURE) incluant des informations sur le moment de la connexion et l'utilisateur. Vue d'ensemble des API : dans la vue d'ensemble des API, on voit toutes les interfaces REST de son application, y compris les codes pour le contrôle de ces API. Base de données : Le serveur JHipster offre également une interface pour les bases de données qui permet de tester et d'établir la connexion. Finalement, à travers l'onglet « Langue », on peut modifier la langue de l'interface de son instance de serveur Spring Boot.



FIGURE 3.19 – Jhipster

3.6 Conclusion

Durant ce chapitre, nous avons décrit l'avant dernière étape du développement celle de l'implémentation. Tout d'abord nous avons présenté l'architecture générale d'une application web. Ensuite, les différentes technologies utilisées. Pour aboutir enfin à l'environnement et les différents outils utilisés.

Chapitre 4

Réalisation et Interfaces

4.1 L'application web "SeQuorTalib"

4.1.1 Les différentes interfaces

Les figures suivantes illustres les différents affichage de l'application web.

Authentification

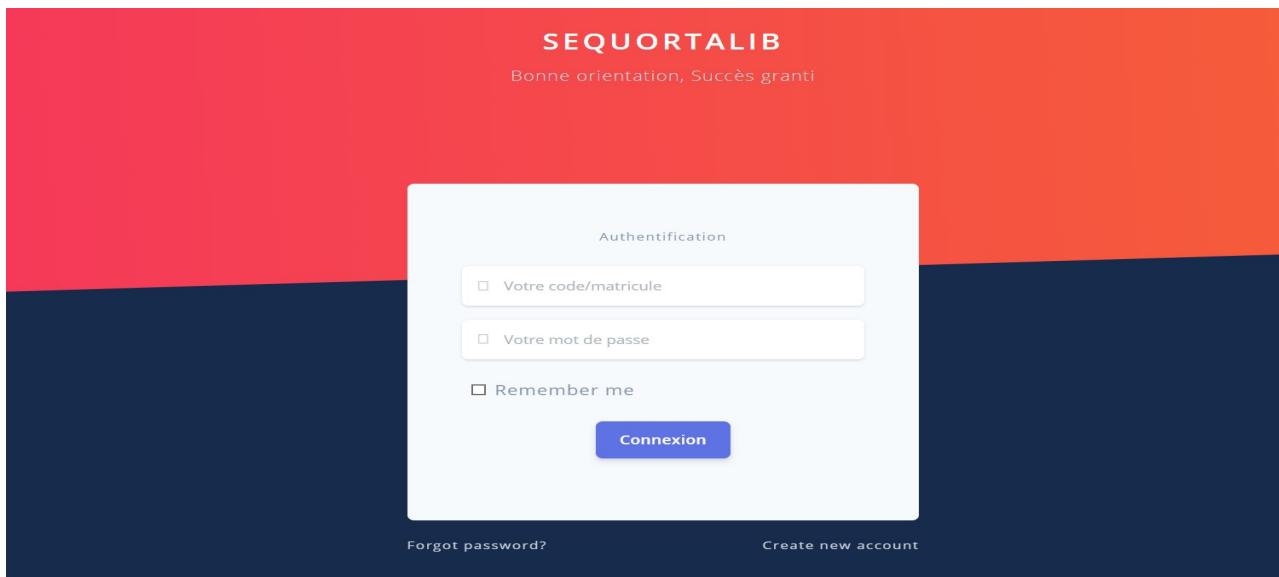


FIGURE 4.1 – Interface authentification

La figure 4.1 représente l'interface qui permet à l'ensemble des utilisateurs de

s'authentifier. Chaque étudiant doit saisir son code d'étudiant, contrairement à l'enseignant il doit saisir son matricule.

Création du compte



FIGURE 4.2 – Création du compte

La figure 4.2 représente l'interface qui permet à l'utilisateur de créer son compte en saisissant le code (étudiant) ou matricule (enseignant), ainsi que son email et le mot de passe qu'il choisit.

Exemple d'un message d'erreur

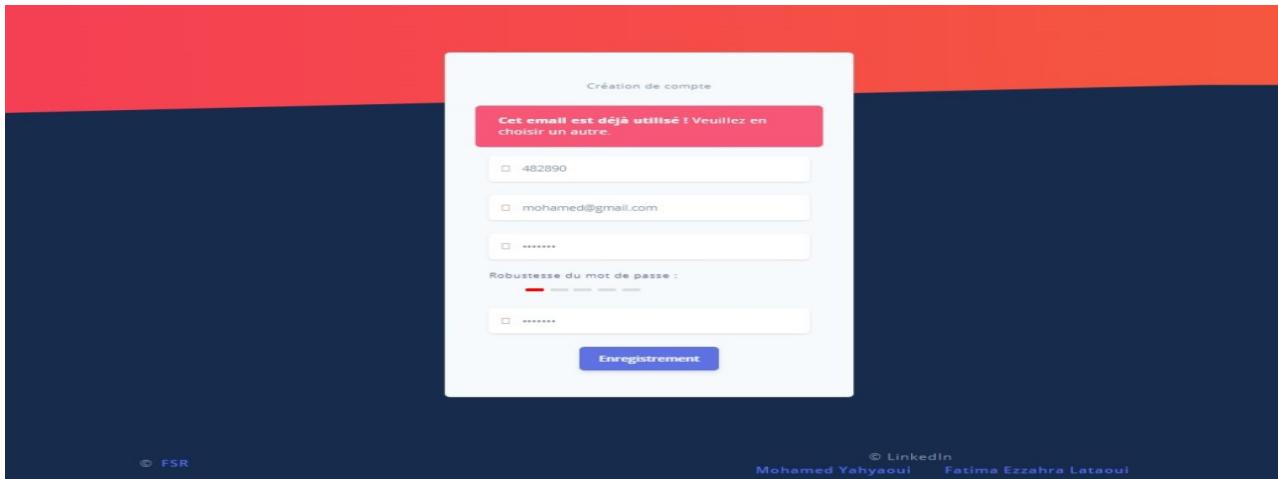


FIGURE 4.3 – Message d’erreur

La figure 4.3 représente la page qui illustre un message d’erreur affiché lorsque l’utilisateur a saisi un email qui existe déjà dans la base de données

Page d'accueil de l'administrateur

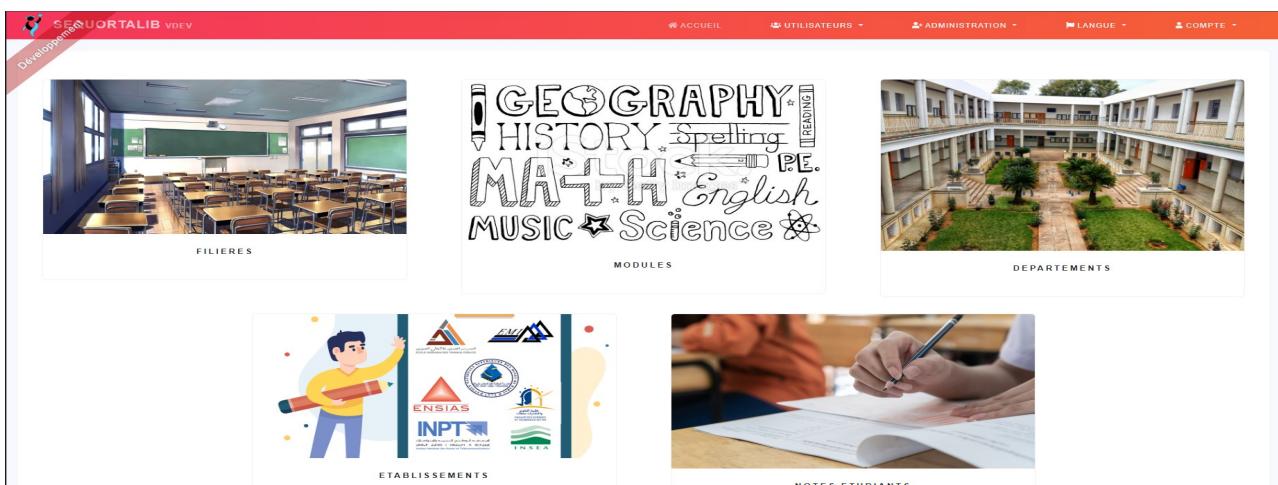
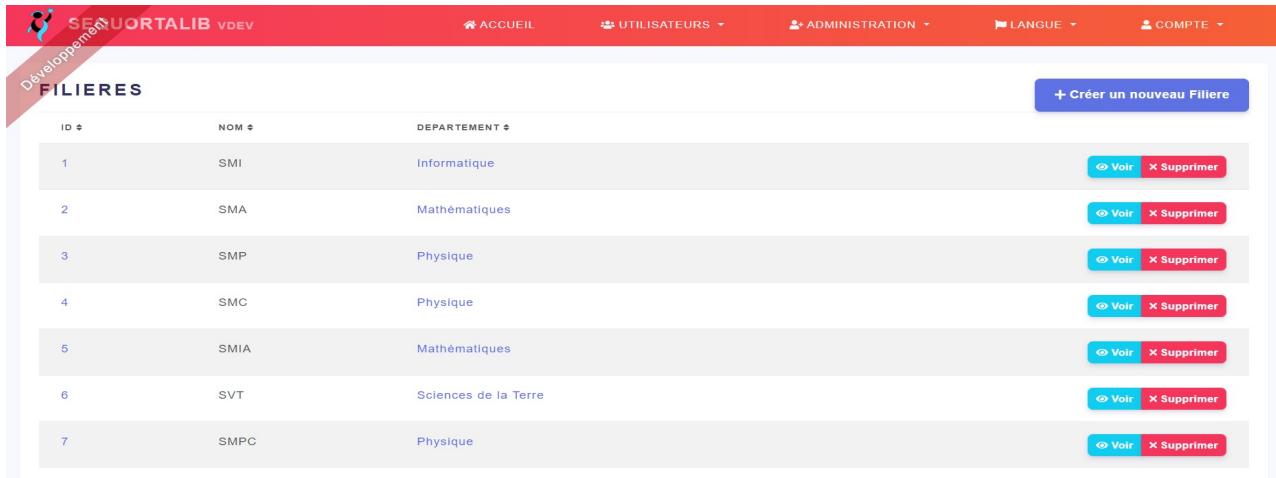


FIGURE 4.4 – Page d’accueil administrateur

La figure 4.4 représente l’interface de la page d’accueil de l’administrateur, ce dernier à un accès aux pages suivantes : page des filières, page des modules, page

des départements, page des établissements, et la page des notes des étudiants.

Liste des filières



ID	NOM	DÉPARTEMENT	
1	SMI	Informatique	Voir Supprimer
2	SMA	Mathématiques	Voir Supprimer
3	SMP	Physique	Voir Supprimer
4	SMC	Physique	Voir Supprimer
5	SMIA	Mathématiques	Voir Supprimer
6	SVT	Sciences de la Terre	Voir Supprimer
7	SMPC	Physique	Voir Supprimer

FIGURE 4.5 – Liste des filières

La figure 4.5 représente l’interface qui permet à l’administrateur d’afficher la liste des filières de la faculté. Il a droit désormais de : voir une filière, supprimer une filière, et/ou créer une nouvelle filière.

Espace des étudiants

The screenshot shows a web application interface for managing student data. At the top, there is a navigation bar with links for ACCUEIL, UTILISATEURS, ADMINISTRATION, LANGUE, and COMPTE. A red ribbon banner on the left says "Développement SEQUORTALIB VDEV". Below the navigation, there are two main sections:

- ÉTUDIANTS DE LA FACULTÉ**: This section contains a large image of a lecture hall filled with students at desks with laptops. Below the image is a table with three rows of student codes:

SMIA	SMPC	SVT
SMI	SMP	SVU
SMA	SMC	SVI
- ANCIENS ÉTUDIANTS DE LA FACULTÉ**: This section contains a large image of a smiling young woman in a graduation cap and gown, with other graduates in the background.

FIGURE 4.6 – Espace étudiant

La figure 4.6 représente la page d'espace d'étudiants permet à l'administrateur d'avoir accès à la liste des : étudiants diplômés ,étudiants inscrit actuellement à la faculté. Il peut aussi ajouter un nouveau étudiant via cette interface.

Les comptes des étudiants

The screenshot shows a table of student accounts. The columns are: ID, CODE/MATRICULE, EMAIL, LANGUE, DROITS, CRÉÉ LE, MODIFIÉ PAR, and MODIFIÉ LE. Each row contains a set of student details and actions:

ID	CODE/MATRICULE	EMAIL	LANGUE	DROITS	CRÉÉ LE	MODIFIÉ PAR	MODIFIÉ LE	Actions
5	1111	salma@gmail.com	fr	ROLE_ETUDIANT	13/09/20 00:03	admin	15/09/20 17:10	<button>Voir</button> <button>Supprimer</button>
7	1212	mohamed@gmail.com	fr	ROLE_ETUDIANT	13/09/20 02:00	admin	15/09/20 17:09	<button>Voir</button> <button>Supprimer</button>
8	3333	qaso@gmail.com	fr	ROLE_ETUDIANT	14/09/20 19:57	admin	14/09/20 20:02	<button>Voir</button> <button>Supprimer</button>
9	3455	aminejamai@gmail.com	fr	ROLE_ETUDIANT	15/09/20 16:57	admin	15/09/20 16:57	<button>Voir</button> <button>Supprimer</button>
10	9000	mohamedhd@gmail.com	en	ROLE_ETUDIANT	15/09/20 17:00	admin	15/09/20 17:00	<button>Voir</button> <button>Supprimer</button>

A message at the bottom center says "Affichage 1 - 8 de 8 items." followed by a set of navigation icons.

FIGURE 4.7 – Les comptes des étudiants

La figure 4.7 représente l'interface qui permet à l'administrateur d'afficher la liste des compte des étudiants, ainsi qu'il peut : voir un compte, supprimer un compte, créer un nouveau compte et voir l'état du compte (activé / désactivé)

Les graphes de suivi



FIGURE 4.8 – Les graphes de suivi

La figure 4.8 représente la page qui contient les graphes des établissements montrant le nombre des étudiants qui les ont intégré par le Deug ou une Licence pour la filière SMI.

Les notes d'un étudiants

SEMESTRE 3 : MOHAMED				
MODULE	NOM	NOTE	VLD	ETAT
M15	PROGRAMMATION I	16.62	VALIDÉ	V.AV
M16	ALGORITHME II	19	VALIDÉ	V.AV
M17	PROBABILITÉ ET STATISTIQUE	14	VALIDÉ	V.AV
M18	TECHNOLOGIE DU WEB	9.89	NON VALIDÉ	V.C.S
M19	ELECTRONIQUE	8	NON VALIDÉ	V.C.S
M20	SYSTÈME D'EXPLOITATION	18.62	VALIDÉ	V.AV

FIGURE 4.9 – Notes d'un étudiant

La figure 4.9 représente l'interface qui contient les notes d'un étudiant, d'un semestre en précisant si le module est validé ou pas ainsi que le type de validation : V.AV : Validé avant rattrapage V.AR : Validé après rattrapage V.CS : Validé par compensation

Page d'accueil d'un enseignant

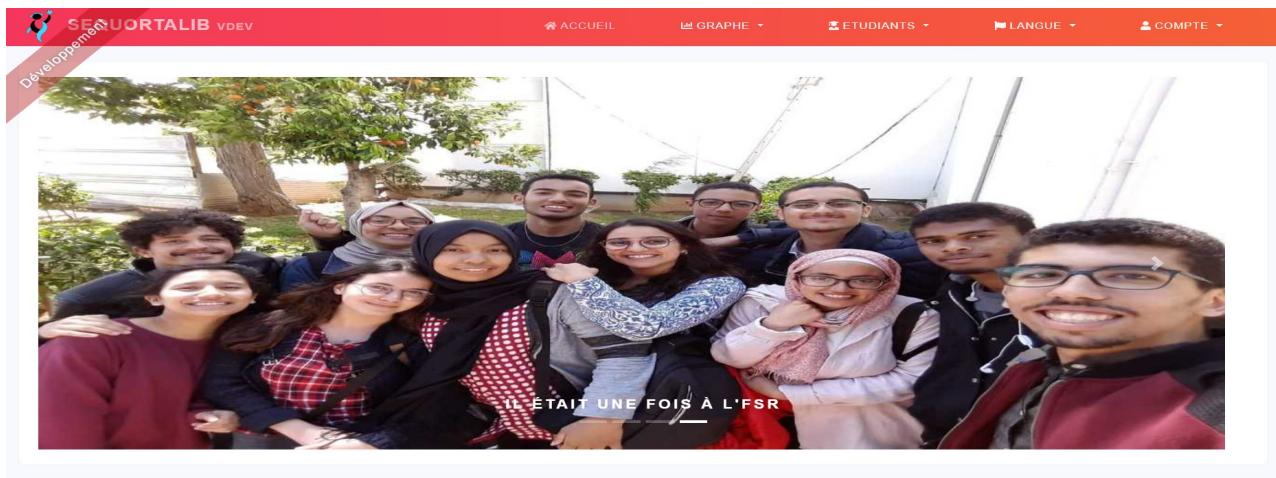
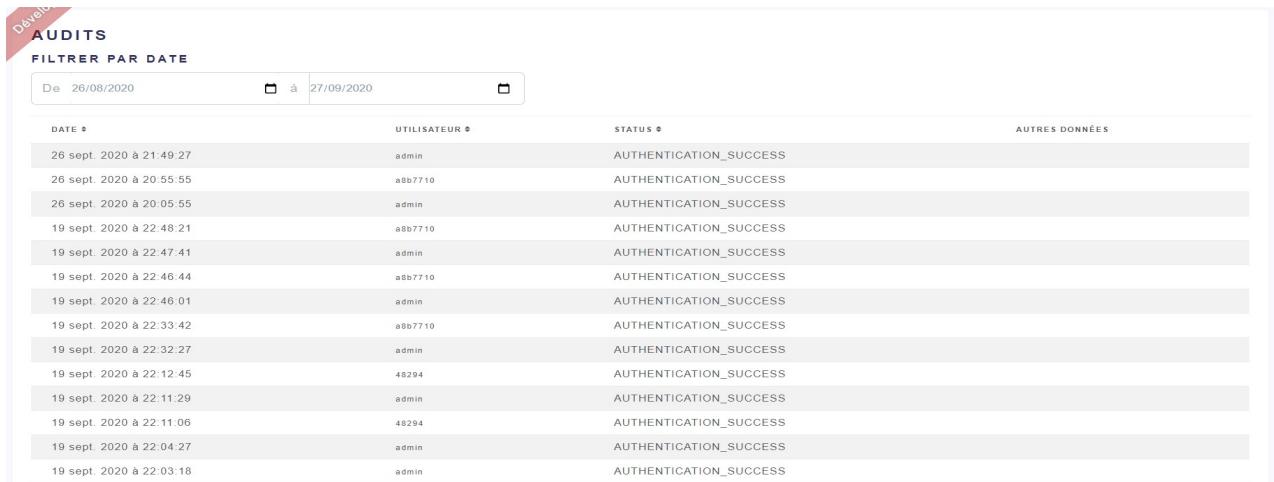


FIGURE 4.10 – Page d'accueil d'un enseignant

La figure 4.10 représente l'interface de la page d'accueil d'un enseignant.

Liste des visiteurs



The screenshot shows a table titled 'AUDITS' with the heading 'FILTRER PAR DATE'. It includes columns for 'DATE', 'UTILISATEUR', 'STATUS', and 'AUTRES DONNÉES'. The table lists 18 rows of audit logs, all showing 'AUTHENTICATION_SUCCESS' status. The logs are timestamped from '26 sept. 2020 à 21:49:27' to '19 sept. 2020 à 22:03:18'. A date range filter at the top is set from '26/08/2020' to '27/09/2020'.

AUDITS			
FILTRER PAR DATE			
De	à	27/09/2020	
26 sept. 2020 à 21:49:27	admin	AUTHENTICATION_SUCCESS	
26 sept. 2020 à 20:55:55	a8b7710	AUTHENTICATION_SUCCESS	
26 sept. 2020 à 20:05:55	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:48:21	a8b7710	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:47:41	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:46:44	a8b7710	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:46:01	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:33:42	a8b7710	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:32:27	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:12:45	48294	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:11:29	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:11:06	48294	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:04:27	admin	AUTHENTICATION_SUCCESS	
19 sept. 2020 à 22:03:18	admin	AUTHENTICATION_SUCCESS	

FIGURE 4.11 – Liste des visiteurs

La figure 4.11 représente la liste des visiteurs de l'application. L'administrateur peut également filtrer(trier) l'affichage selon la date.

Métrique et informations machine

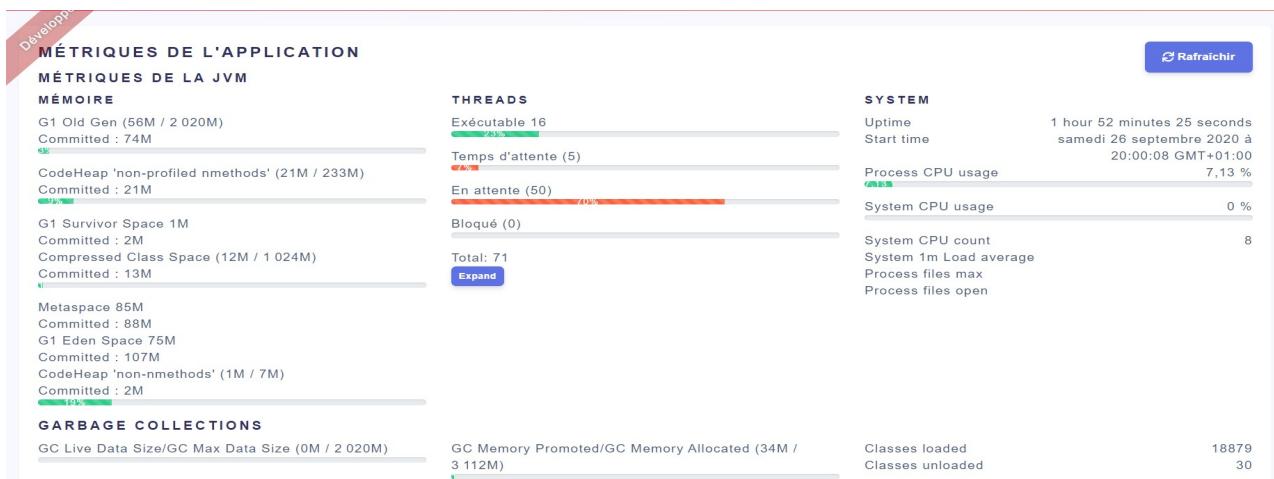


FIGURE 4.12 – Métrique et informations machine

La figure 4.12 représente des informations globales sur l'état de la machine : la mémoire, le processeur, le temps d'exécution...

4.1.2 L'affichage de la base de données sous PhpMyAdmin

La table USER

id	login	password_hash	first_name	last_name	email
1	system	\$2a\$10\$mE.qmcV0mFU5NcKh73Tz.x4uel/.bDWbj0T1BYyqP4...	System	System	system@localhost
2	anonymoususer	\$2a\$10\$j8S5d7Sr7.8VTOYNviDPOeWX8KcYILUVJBsYV83Y5Nt...	Anonymous	User	anonymous@localhost
3	admin	\$2a\$10\$gSAhZrxMllrbgj/kkK9UceBPpChGWJA7SYlb1Mqo.n5...	Administrator	Administrator	admin@localhost
4	user	\$2a\$10\$VEjxo0jq2YG9Rbk2HmX9S.k1uZBGYUHdUcid3g/vfiE...	User	User	user@localhost
5	1111	\$2a\$10\$EMxsB0El/cuecoE8nkoU4OwX.AdR/lkbY5QbjbBBe0C...	Hamouchi	Salma	salma@gmail.com
7	1212	\$2a\$10\$Lny6DP8YEdng1048gSL.veBuozmKKd3MeC5wMCXp0Rp...	yahyaoui	Mohamed	mohamed@gmail.com
8	3333	\$2a\$10\$Hi4BA8FNjpAnraBE99RNuepqXfGPR.kG5uK59GI0WiP...	NULL	NULL	qaso@gmail.com
9	3455	\$2a\$10\$iRVwgIY2nQrt0PjvDR7CJOcVHmfK501.5bV0UflSezp...	Jamai	Mohamed amine	aminejamai@gmail.com
10	9000	\$2a\$10\$TEMggIM/NdroW.gBTHftzOPfqqaB3iwAWMgHGx5ClkO...	Hadani	Mohamed	mohamedhd@gmail.com
11	482890	\$2a\$10\$218nMktzGxOrCbYL1sbgr..fUva9jv.W/D7yq1cWibo...	NULL	NULL	mohamed2000@gmail.com

FIGURE 4.13 – Table user

La table étudiant

	<input type="checkbox"/>	<input type="checkbox"/>	Éditer	Copier	Supprimer	id	tel	cin	etat	datenaissance	semestre	section	promo	etablissement_id	filiere_id
	<input type="checkbox"/>	<input type="checkbox"/>				5	645231223	AA23453	NONDIPLOME	1998-09-01	5	A	NULL	NULL	4
	<input type="checkbox"/>	<input type="checkbox"/>				6	6341223	AA89900	DIPLOME	1997-10-01	NULL	NULL	2019	4	NULL
	<input type="checkbox"/>	<input type="checkbox"/>				7	6452367	AA12234	DIPLOME	1999-08-31	NULL	NULL	2018	16	NULL
	<input type="checkbox"/>	<input type="checkbox"/>				8	64512349	AA23341	DIPLOME	1983-08-30	NULL	NULL	2016	9	NULL

↑ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

FIGURE 4.14 – Table étudiant

La table filière

	<input type="checkbox"/>	<input type="checkbox"/>	Éditer	Copier	Supprimer	id	nom	departement_id
	<input type="checkbox"/>	<input type="checkbox"/>				1	SMI	5
	<input type="checkbox"/>	<input type="checkbox"/>				2	SMA	4
	<input type="checkbox"/>	<input type="checkbox"/>				3	SMP	3
	<input type="checkbox"/>	<input type="checkbox"/>				4	SMC	3
	<input type="checkbox"/>	<input type="checkbox"/>				5	SMIA	4
	<input type="checkbox"/>	<input type="checkbox"/>				6	SVT	2
	<input type="checkbox"/>	<input type="checkbox"/>				7	SMPC	3

FIGURE 4.15 – Table filière

La table département

The screenshot shows a table titled "département" with columns "id" and "nom". The data is as follows:

			id	nom
<input type="checkbox"/>	Éditer	Copier	1	Biologie
<input type="checkbox"/>	Éditer	Copier	2	Sciences de la Terre
<input type="checkbox"/>	Éditer	Copier	3	Physique
<input type="checkbox"/>	Éditer	Copier	4	Mathématiques
<input type="checkbox"/>	Éditer	Copier	5	Informatique

At the bottom, there are buttons for "Tout cocher" (Select all), "Avec la sélection : " (With selection :), "Éditer" (Edit), and "Copier" (Copy).

FIGURE 4.16 – Table département

4.2 Conclusion

Durant ce chapitre, on a décrit la dernière étape du développement qui est celle de la réalisation et la mise en œuvre du projet. D'abord on a présenté les outils utilisés, puis l'architecture logicielle de l'application en présentant le modèle en couches. Et pour finir on a donné quelques captures d'écran de l'application réalisée.

Conclusion et perspectives

Avant tout, SeQuorTalib , repose sur deux aspects primordiaux. Le premier consiste à faire un suivi des étudiants diplômés. En effet, ils ont eu leurs licences pour faire un Master ou intégrer un cycle d'Ingénieur dans un autre établissement. En d'autres termes réaliser un historique de leur parcours au sein de la faculté. Puis le deuxième qui est bien évidemment la réalisation d'un suivi des étudiants actuellement inscrits en assimilant une certaine gestion cohérente et convenable des notes. En outre, parmi les fonctionnalités globales de l'application, nous avons la consultation de plusieurs et différents graphes, l'accès aux notes ainsi que la gestion des établissements, modules, filières, étudiants et établissement.

Cette application réalise un suivi bien exacte des étudiants désormais diplômés de la faculté des sciences de Rabat. Cependant, elle garantit une organisation d'une communication indirecte entre l'enseignant et ses étudiants actuellement inscrits au sein de l'établissement sans se rendre aux autres méthodes plus compliquées qui peuvent coûter plus de temps, ainsi qu'assurer un espace privé de chaque étudiant contenant ses informations personnelles et ses notes présentées dans des graphes. D'ailleurs, nous pouvons bien l'améliorer en spécifiant les graphes des filières des établissements, au lieu de se limiter seulement sur ceux de ces derniers d'une manière générale. En outre, et en se basant sur la filière et le semestre de chaque étudiant, nous pouvons bien proposer des cours certifiés offerts par les plateformes, par exemple Udemy, EDX et coursera cette dernière qui collabore avec l'université de Rabat. Et finalement, donner une probabilité à l'étudiant d'avoir une certaine mention de son diplôme en cours de préparation ou d'avoir la possibilité de passer le concours de certaines grandes écoles. Pourraient bien lui donner une vision sur son rythme et ses avancements et lui permettent aussi d'organiser sa façon de travailler.

En ce qui concerne le plan de travail, nous avons en premier lieu effectué une phase d'étude des outils d'existant. Ensuite, nous avons spécifié notre application pour identifier ses fonctionnalités. Puis, nous avons procédé à la conception ainsi qu'aux choix des technologies pour sa réalisation.

De surcroît, cette application est jugée étant qu'une expérience agréable, dans laquelle nous avons pu acquérir des compétences et des connaissances importantes et des langages de programmation indispensables pour la carrière d'un développeur Web. D'une part, le Java EE et ses frameworks : Spring, AngularJs et Hibernate. D'autre part, l'outil Jhipster, ce dernier demeure parfait, car nous a beaucoup aidé pour réussir ce projet afin de garantir un meilleur résultat sans trop de complications.

Bibliographie

Bibliographie

- [1] Martinal Banon, JPA et Java Hibernate - Apprenez le mapping objet-relationnel (ORM) avec Java – 8 février 2017
- [2] Sébastien Ollivier, AngularJS - Développez aujourd’hui les applications web de demain - 11 février 2015
- [3] Julien Dubois, Jean-Philippe, Retaillé, Thierry Templier et Rod Johnson, Spring par la pratique : Mieux développer ses applications Java/J2EE avec Spring, Hibernate, Struts, Ajax... - 28 avril 2006
- [4] R. PAWLAK, J.-P. RETAILLÉ, L. SEINTURIER. – Programmation orientée aspect pour Java/J2EE. N°11408, 2004.
- [5] L. MAESANO, C. BERNARD, X. LEGALLES. – Services Web en J2EE et .NET. N°11067, 2003.
- [6] <https://makina-corpus.com/blog/metier/2017/premiers-pas-avec-rxjs-dans-angular> - Makina Corpus - 22/09/2017 - visité le 03/02/2020
- [7] www.udemy.com/course/full-stack-application-development-with-spring-boot-and-angular - in28minutes - mars 2018 - visité le 21/04/2020
- [8] <https://blog.openclassrooms.com/2020/06/20/les-10-metiers-les-plus-recherches-dans-le-numerique/> - Nora Leon - 20/06/2020 - visité le 10/09/2020
- [9] <https://stackoverflow.com/questions/47381606/npm-install-command-please-explain> - avril 2018 - visité le 02/03/2020
- [10] <https://github.com/jhipster/generator-jhipster/issues/7843> - 23/06/2018 - visité le 12/06/2020
- [11] <https://www.chartjs.org/docs/latest/getting-started/usage.html> - visité le 12/05/2020