

REMERCIEMENTS

Avant tout développement sur cette expérience, il apparaît de commencer ce rapport par des remerciements à ceux qui nous ont beaucoup appris au cours de ce projet.

A nos encadrements

Nous exprimons toute notre gratitude et nos remerciements à Monsieur Ali Bekri, Monsieur Ali Oubelkacem qui n'ont pas cessé de nous encourager pendant la durée du projet et pour l'intérêt qu'ils ont porté à notre travail. Nous les remercions également pour leurs encouragements et pour leurs conseils.

Nous tenons à remercier également, Mme Chaymaa Lamini de nous avoir incité à travailler en nous faisant part de ses expériences et de ses compétences.

Aux membres du jury

Messieurs les jurys, vous nous faites un grand honneur en acceptant de juger ce travail.

Nous adressons le remerciement à tous les enseignants de notre Bachelor pour leurs qualités scientifiques et pédagogiques.

Nous tenons à remercier chaleureusement, tous nos proches et tous ceux qui, de près ou de loin, comment nous ont apporté leurs sollicitudes pour accomplir ce travail.

RESUME

Le present rapport constitue le resultat d'un travail realise dans le cadre du projet S4 de notre formation BGI. Notre projet consiste dans l'étude, la conception et la réalisation d'une application web destinée à la gestion administrative du Bachelor.

Ce projet est particulièrement orienté développement avec l'environnement JEE, qui permet aux différents utilisateurs (Administrateur-professeur-etudiant) de faciliter et réaliser les différentes tâches reliées à l'administration du Bachelor. Ce qui est réalisable à travers les différentes interfaces proposées par notre application. Ces fonctionnalités sont les objectifs majeurs de notre application.

A travers ce document, nous allons décrire en détail les différentes étapes de réalisation de ce projet.

LISTE DES FIGURES

Figure 1: Diagramme de GANTT

Figure 2: Diagramme de cas d'utilisation

Figure 3: Diagramme de séquence pour l'inscription des étudiants

Figure 4: Diagramme de séquence pour la réservation d'une salle

Figure 5: Diagramme de séquence pour la suppression d'une absence

Figure 6: Diagramme de classes

Figure 7: Architecture logicielle du projet

TABLE DE MATIERE

I.	INTRODUCTION.....	6
II.	PRESENTATION GENERALE	6
1.	Problématique:	6
2.	Objectifs:.....	7
III.	ETUDE PRELIMILAIRE.....	7
1.	Spécification des fonctionnalités attendues:.....	7
2.	Planification:	7
IV.	ANALYSE ET CONCEPTION	8
1.	Identification des acteurs	8
2.	Principale fonctionnalité du système	9
3.	Vue globale du système.....	9
4.	Diagramme de cas d'utilisation :.....	10
5.	Description textuelle de quelques cas d'utilisation:	10
6.	Diagrammes de séquences:.....	14
7.	Diagramme des classes.....	18
V.	OUTILS DE DEVELOPPEMENT	18
VI.	MISE EN ŒUVRE DU PROJET.....	23
VII.	DEMONSTRATION.....	25
VIII.	CONCLUSION ET PERSPECTIVES.....	34
IX.	BIBLIOGRAPHIE.....	35

I. INTRODUCTION

Dans le cadre de notre formation de notre seconde année du cycle Génie Informatique au Bachelor de l'Université Moulay Ismaïl, il nous'est proposé un projet de 3 mois nous permettant de mettre en pratique nos connaissances théoriques au travers la conception et le développement d'une application destinée à la gestion administrative du Bachelor.

Plusieurs outils ont été utilisés pour permettre le développement des différents modules réalisés au cours du projet.

Le résultat de ce travail est une application web . Il dispose de plusieurs modules répondant chacun à des besoins particuliers : Interface admin, Interface professeur, Interface étudiant. Ce rapport se décompose en cinq parties :

- **Contexte général du projet** : comprenant la présentation générale du projet, la problématique et ses objectifs.
- **Etude préliminaire** : comprenant les besoin fonctionnels, présentation du cahier de charges et la planification en utilisant le diagramme de Gantt.
- **Analyse et conception** : comprenant une étude conceptuelle réalisé par le langage de modélisation UML. Nous sommes seulement limites à trois modules, le diagramme cas d'utilisation et ses descriptions textuelles correspondantes, les diagrammes de séquence et les diagrammes de classes.
- **Environnement et outils** : comprenant les outils utilisés pour le développement du système.
- **Mise en œuvre de projet** : comprenant l'architecture de l'application et le guide d'utilisation de l'application

II. PRESENTATION GENERALE

1. Problématique:

La formation Bachelor à Meknès suscite de plus en plus d'engouement de la part des nouveaux bacheliers au niveau nationale. L'accroissement de la demande vis-à-vis de cette formation donne un travail de plus en plus lourds à l'administration. L'informatisation de la gestion administrative est une bonne solution pour faciliter, entre autres, les tâches du recensement des absences des étudiants inscrits dans les différentes filières et semestres, et pour le suivi des appointements des enseignants.

2. Objectifs:

Cette application développée a pour but non seulement de mettre à disposition des personnes s'occupant de l'administration de la formation Bachelor, des outils adéquats pour mener à bien ces tâches quotidiennes, mais aussi permettre d'autres outils aux étudiants et aux professeurs pour une communication plus rapide et effective avec l'administration.

III. ETUDE PRELIMILAIRE

1. Spécification des fonctionnalités attendues:

Notre application va introduire les traitements suivants :

- Pour la gestion des étudiants, l'administrateur peut ajouter, modifier et supprimer des étudiants .
- Gérer les réservations et les disponibilités des salles.
- Ajouter les emplois du temps de chaque filière, les visualiser ou les imprimer.
- Ajouter les informations concernant un professeur .
- L'administrateur peut ajouter, modifier, consulter et supprimer les absences des étudiants.
- Gestion des sessions d'administrateur, professeur et étudiant.

2. Planification:

Le Diagramme de GANTT est un outil de gestion de projet, et l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. Un diagramme de Gantt répertorie toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées.

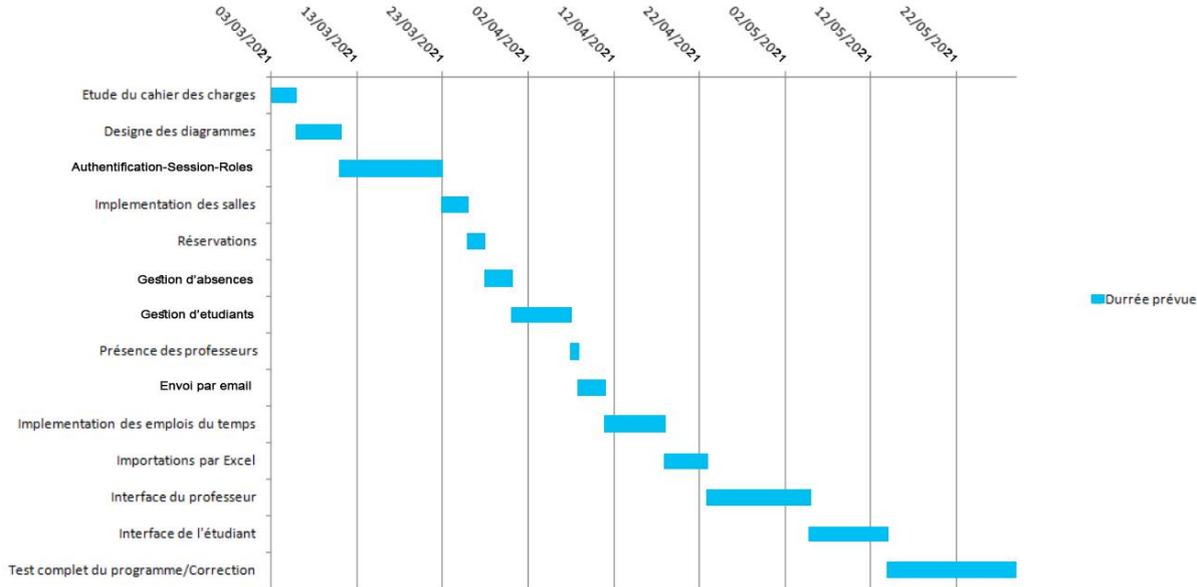


Figure 1: Diagramme de GANTT

IV. ANALYSE ET CONCEPTION

1. Identification des acteurs

Les Acteurs	Les rôles
Administrateur	1-Effectuer Inscriptions administratives 2-Création des comptes d'authentification 3-Ajout des données (étudiants-professeurs-salles-emplois du temps-modules-filières-des absences) 4-Modification des données (étudiants-professeurs-salles-emplois du temps-modules-filières-des absences) 5-Suppression des données (étudiants-professeurs-salles-emplois du temps-modules-filières-des absences) 6-Génération des feuilles d'absences 7-Envoie des emails aux parents 8-Consultation des données (heures d'absences-listes des étudiants) 9-Impression des emplois du temps 10- L'authentification
Professeur	1-Ajout de l'absence et notation de sa présence 2-Réservation des salles 3-Consultation des données (les réservations-propre emploi du temps-les salles) 4-Consultation fiche d'appointment 5-Impression des emplois 6-L'authentification
Etudiant	1-Consultation d'emploi et d'absences 2-L'impression des emplois 3-L'authentification

2. Principale fonctionnalité du système

La principale fonctionnalité de ce système est la manipulation d'une base de données des branches du Bachelor à travers une application web. Plus précisément l'application nous permet de créer, modifier, ou supprimer des étudiant, des professeurs, des absences, des salles, des réservations ou des emplois du temps.

3. Vue globale du système

Le système gère les droits et les autorisations des types d'utilisateurs dans le même cadre de la gestion d'administration. Nous avons des utilisateurs qui ont tous les droits (les administrateurs), et des utilisateurs qui ont des droits limités (professeurs et étudiants).

4. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est une vue statique qui montre les fonctionnalités d'un système. Ces cas d'utilisations mettent en œuvre les acteurs du système dans leurs relations.

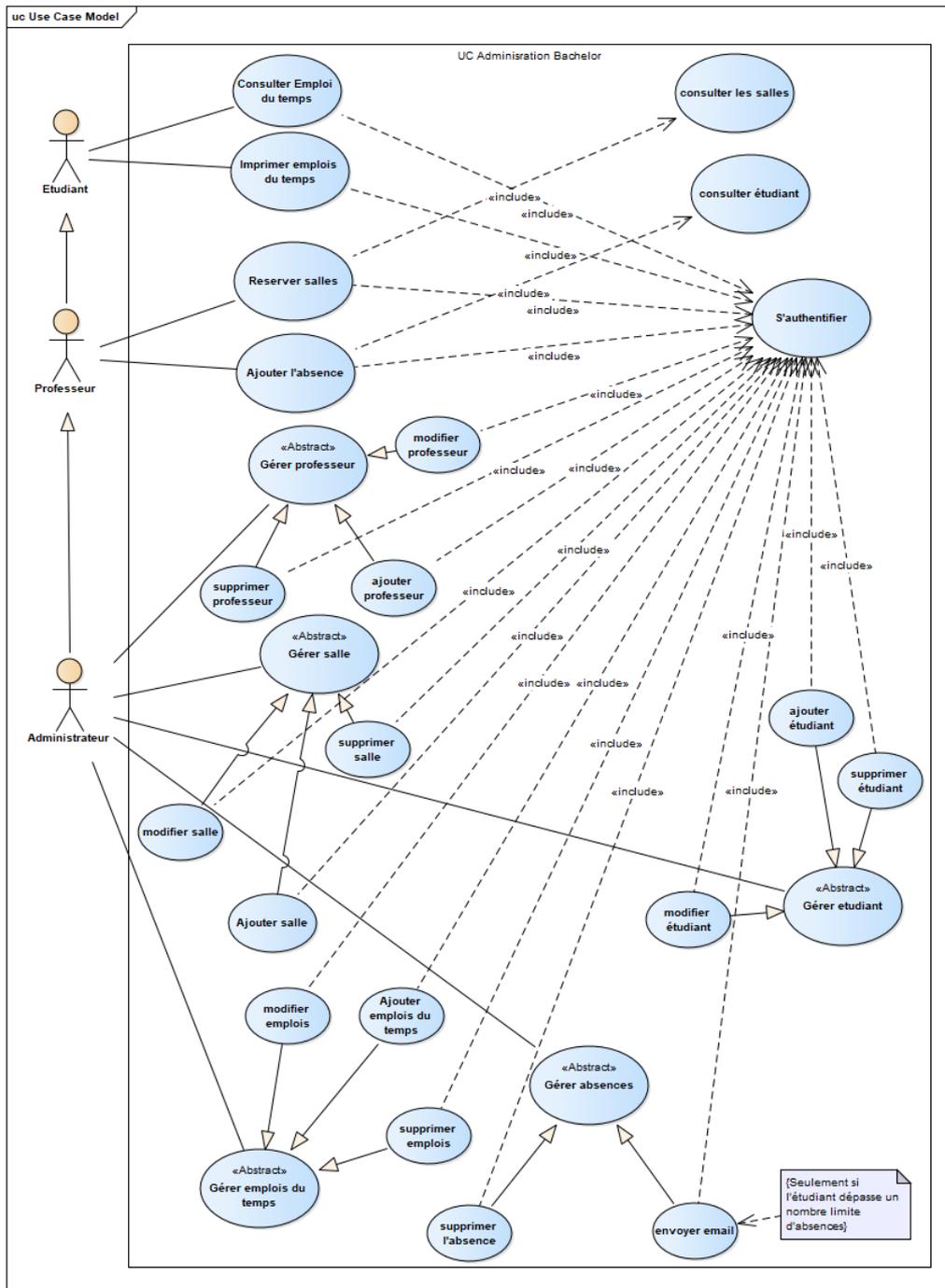


Figure 2: Diagramme de cas d'utilisation

5. Description textuelle de quelques cas d'utilisation:

Nom Du CU	<u>Authentification</u>
Acteurs	L'utilisateur de l'application (Administrateur, professeur, étudiant).
Résumé	L'utilisateur devra saisir ses données de connexion afin d'accéder à l'application web.
Préconditions	L'utilisateur doit être créé dans la base de données et connaître ses identifiants.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur ouvre l'application 2. Le système affiche la page d'authentification. 3. L'acteur saisit le login et le mot de passe. 4. Le système vérifie l'existence des données dans la base de données. 5. Le système affiche la page d'accueil.
Scénario Alternatif	<p>A. Erreur d'authentification : login ou mot de passe non valide. - Cet enchaînement démarre au point 4.</p> <p>5. Le système affiche un message d'erreur. Le scénario reprend au point 2.</p> <p>B. Champs obligatoires vides. - Cet enchaînement démarre au point 4. Le scénario reprend au point 2.</p>
Scénario d'exception	<ol style="list-style-type: none"> 1. L'utilisateur n'a pas saisi les bons identifiants. 2. Le système renvoie un message d'erreur. 3. L'utilisateur n'existe pas dans la base de données.
Postcondition	Ouverture de l'espace utilisateur.

Nom Du CU	Réservation des salles
Acteurs	Professeur
Résumé	Afin de faire un cours ou un TP. Chaque professeur a le droit de réserver une salle à condition qu'elle soit valable
Préconditions	Le professeur s'authentifie à l'application Existence des salles vides
Scénario nominal	1-Le professeur clique sur consulter les salles 2-Le système affiche les salles disponibles avec leur descriptions 3-Le professeur choisi une salle pendant une date et un créneau bien définit. 4-Le système consulte la base de données 5-Le professeur confirme sa réservation
Scénario Alternative	
Scénario d'exception	E1 : le choix de la salle dans un temps définie est impossible L'enchainement E1 démarre au point 3 du scénario nominal 4-Le système consulte la base de données 5-Une page d'erreur s'affiche impossibilité de réserver cette salle
Postcondition	Salle réservée Salle réservée est ajoutée à la base de données

Nom Du CU	Ajouter l'absence et marque la présence
Acteurs	Professeur
Résumé	Le système fournit aux professeurs une interface pour ajouter les absences ainsi qu'à marquer leur présence selon la date et créneau et matière enseigné
Préconditions	Le professeur s'authentifie à l'application
Scénario nominal	<ol style="list-style-type: none"> 1-Le professeur clique sur ajouter absence 2-Le système affiche une interface des étudiants 3-Le professeur marque les absences des étudiants et sa présence 4-Le professeur confirme la tache 5-L'absence est ajoutée à la base de données
Scénario Alternative	
Scénario d'exception	
Postcondition	L'absence est ajoutée à la base de données

Nom Du CU	Effectuer inscription administrative
Acteurs	Administrateur
Résumé	L'administrateur doit effectuer une inscription administrative pour chaque étudiant admit au Bachelor selon la filière, en introduisant ses informations personnelles
Préconditions	L'administrateur s'authentifie à l'application L'étudiant à inscrire, doit être déjà admit au Bachelor
Scénario nominal	1-L'adminastrateur sélectionne l'inscription administrative 2-Le système affiche un formulaire à remplir 3-L'adminstrateur saisie les informations personnelles de l'étudiant et ses documents administratifs 4- Le système consulte la base de données 5- L'administrateur confirme l'inscription
Scénario Alternatif	A1 : l'administrateur peut importer un fichier Excel contenant les informations des étudiants A1 commence à l'étape 3 du Scénario nominal. L'enchaînement continue nominalement.
Scénario d'exception	E1 :la redondance des données d'un des étudiants inscrits E1 se trouve dans la 4eme étape du scénario nominal Le système affiche une erreur que l'étudiant déjà existe. E2 : l'administrateur n'a pas entrer l'une des importantes informations. Le système force l'administrateur à remplir ces champs.
Postcondition	Etudiant est inscrits administrativement et ajouté à la base de données

6. Diagrammes de séquences:

Qui sont une représentation temporelle des objets et de leurs intersections.

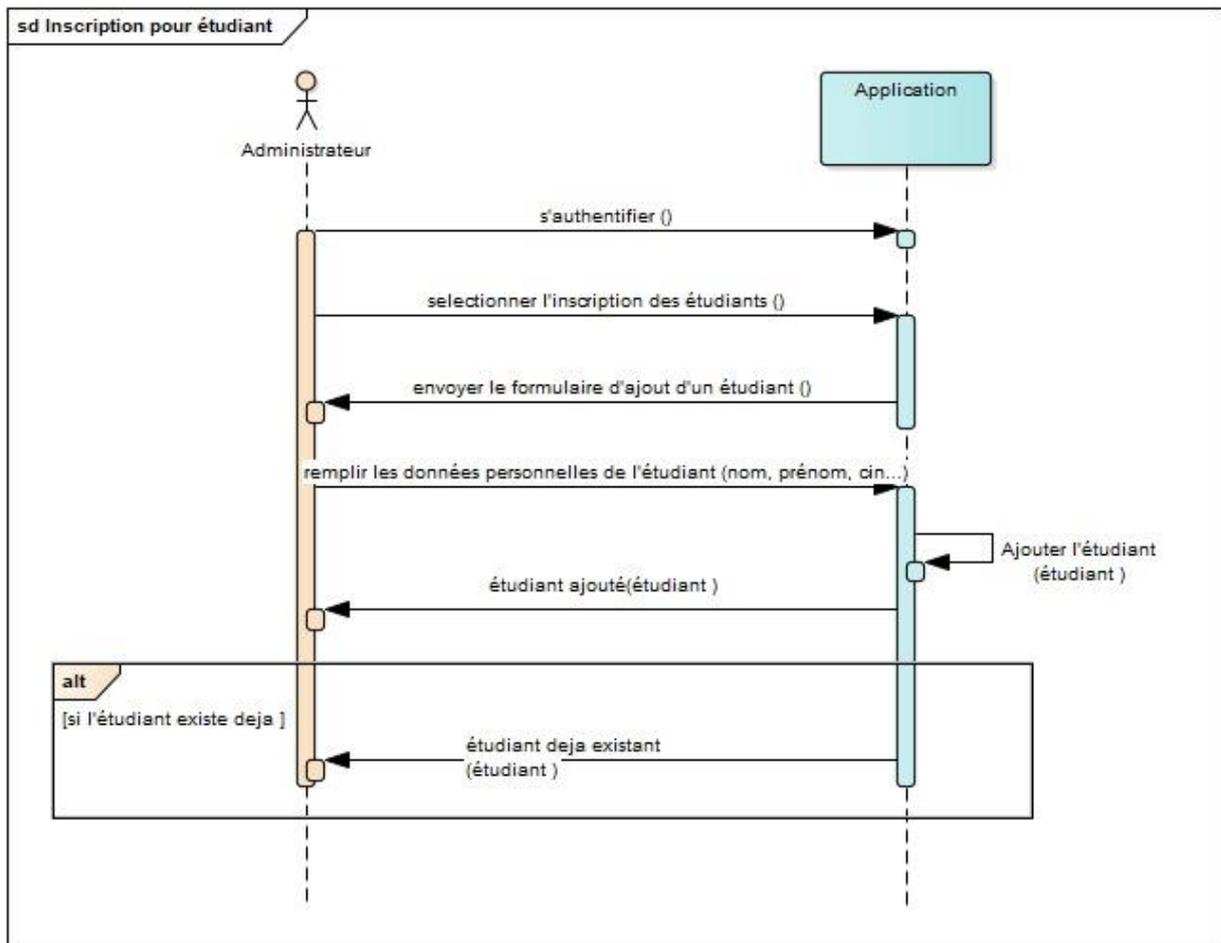


Figure 3: Diagramme de séquence pour l'inscription des étudiants

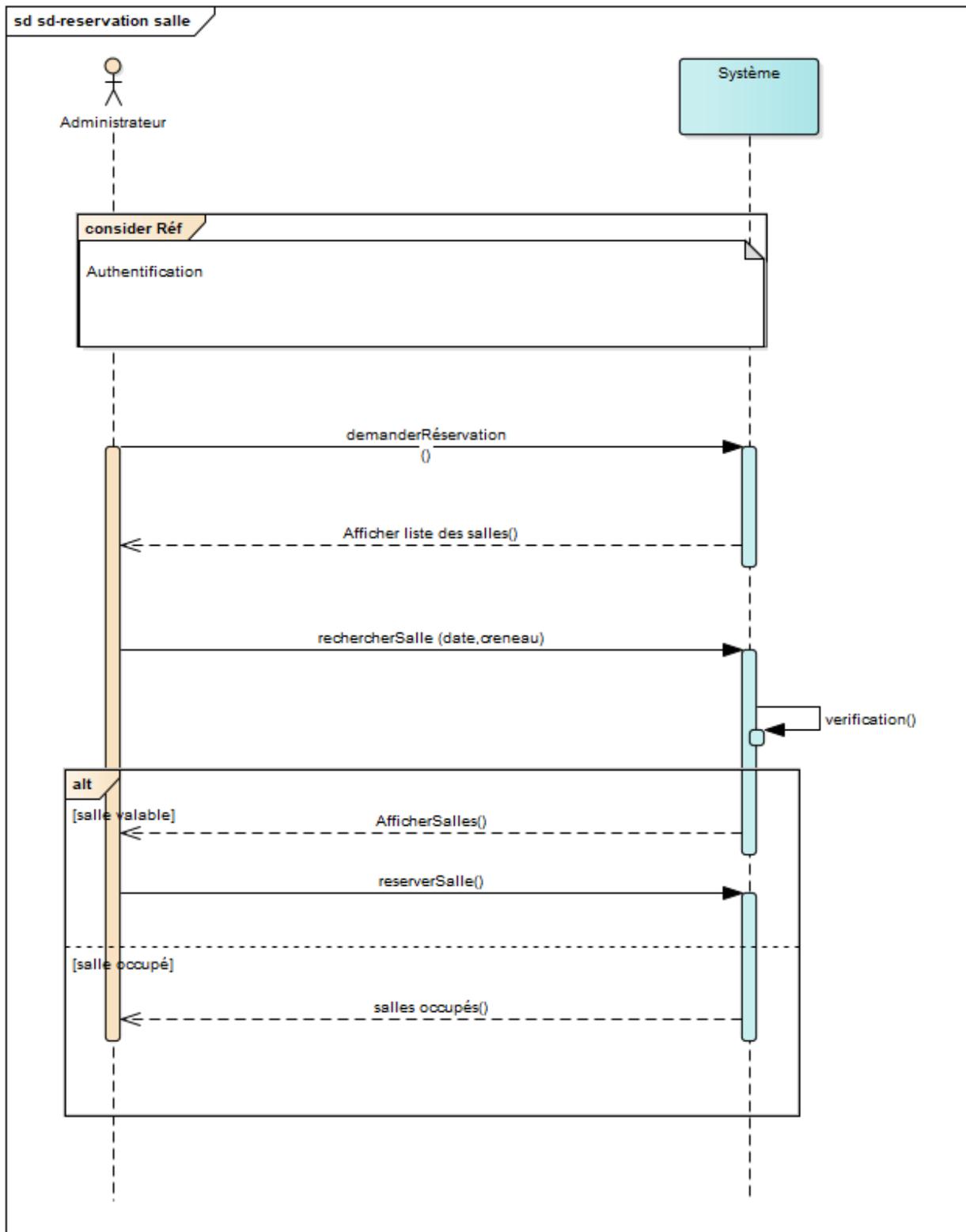


Figure 4: Diagramme de séquence pour la réservation d'une salle

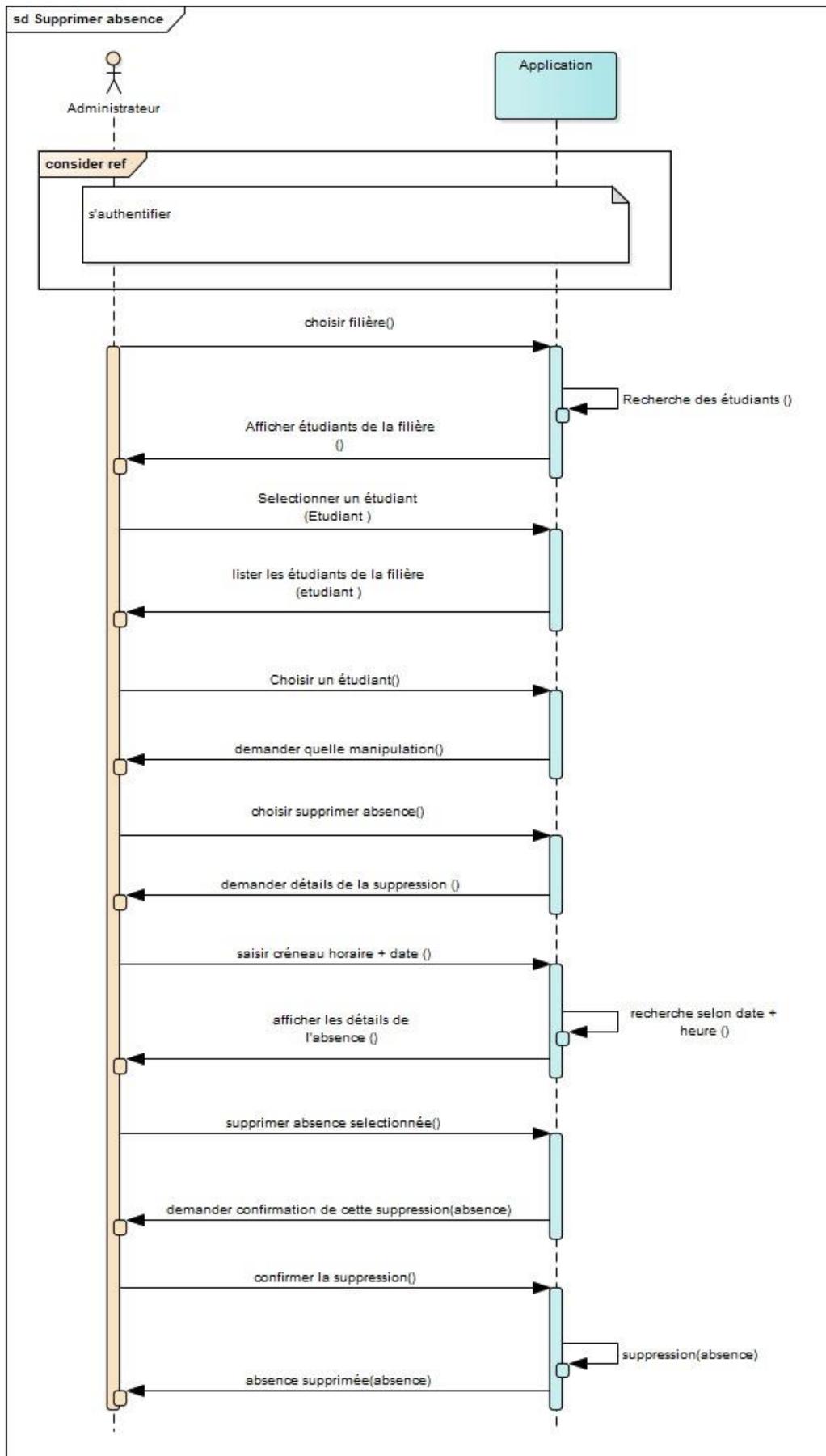


Figure 5: Diagramme de séquence pour la suppression d'une absence

7. Diagramme des classes

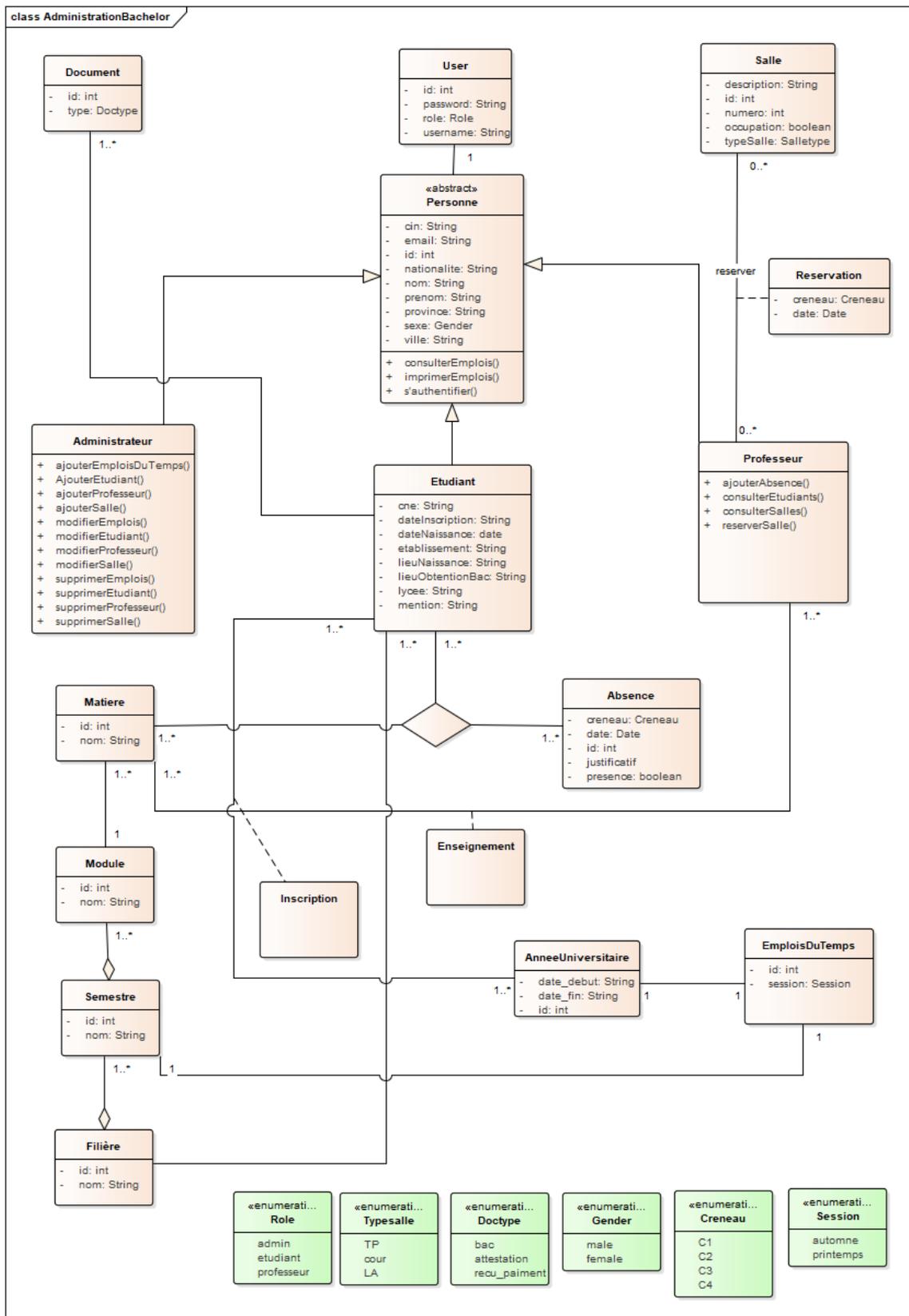


Figure 6: Diagramme de classes

V. OUTILS DE DEVELOPPEMENT

- **Java EE**, Java Entreprise Edition ou rebaptisé Jakarta EE, est un Framework à la fois riche (basé sur la bibliothèque Java SE et possède de nombreuses API), ouvert (la communauté Java Community Process) et dédié au développement, au déploiement et à l'exécution d'application Internet modernes (nécessaires aux entreprises). Il s'agit d'une plate-forme fortement orienté serveur pour le développement et l'exécution d'application distribuées, cela dit bien sûre avec le langage de programmation Java.



- **Hibernate** est un Framework ORM (Object Relational Mapping) pour Java dans les bases de données relationnelles. Ce Framework est capable de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa. en utilisant son propre langage de programmation qui est basé sur le langage SQL.



- **Maven** est un outil de gestion de projet qui comprend un modèle objet pour définir un projet, un ensemble de standards, un cycle de vie, et un système de gestion des dépendances. Il embarque aussi la logique nécessaire à l'exécution d'actions pour des phases bien définies de ce cycle de vie, par le biais de plugins. Lorsque vous utilisez Maven, vous décrivez votre projet selon un modèle objet de projet clair, Maven peut alors lui appliquer la logique transverse d'un ensemble de plugins (partagés ou spécifiques).



- Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation .



- **MySQL**, le plus populaire des serveurs de bases de données SQL Open Source, est développé, distribué et supporté par MySQL AB. MySQL AB est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL.



- Apache Tomcat est un conteneur libre de Servlet Java EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP (web).



- **HTML** est un langage de description de document utilisé sur Internet pour faire des pages Web. Son sigle signifie « HyperText Markup Language » en anglais, littéralement « langage de marquage hypertexte ». Le balisage HTML est incorporé dans le texte du document et est interprété par un navigateur Web.



- Les feuilles de style en cascade, généralement appelées **CSS** de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium.



- Javascript, créé à l'origine par Netscape, ce langage de programmation est conçu pour traiter localement des événements provoqués par le lecteur (par exemple, lorsque le lecteur fait glisser la souris sur une zone de texte, cette dernière change de couleur). C'est un langage interprété, c'est-à-dire que le texte contenant le programme est analysé au fur et à mesure par l'interprète, partie intégrante du browser, qui va exécuter les instructions. Ce langage a fait l'objet d'une normalisation sous le nom de ECMAScript.



- **jQuery** n'est en fait qu'un seul et unique fichier JavaScript téléchargeable sur le web. C'est une énorme bibliothèque de fonctions JavaScript qui ont été écrites et regroupées pour plus de simplicité. Cependant, jQuery est plus qu'une simple librairie même s'il en a les traits majeurs, il va vraiment vous faire coder en JavaScript d'une nouvelle manière. Et ceci à tel point qu'il est tout à fait possible de considérer jQuery comme un langage un peu à part, puisqu'il s'agit vraiment d'une redécouverte totale de JavaScript.



- **Bootstrap** est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



Bootstrap

- **Enterprise Architect:** L'UML s'est imposé comme le standard en matière de langage de modélisation des systèmes informatiques. De l'expression des besoins au déploiement en passant par la conception, utiliser ou tout simplement comprendre UML est un incontournable de tout acteur du SI. Le logiciel Enterprise Architect de Sparx System permet de réaliser tous les diagrammes UML 2. Bien plus qu'un simple modèleur UML, il permet de mettre en œuvre toutes les phases et disciplines d'un projet.



- **Git** est un logiciel de gestion de versions (Version Control System) qui suit l'évolution des fichiers sources et garde les anciennes versions de chacun d'eux sans rien écraser. Cela permet de retrouver les différentes versions d'un fichier ou d'un lot de fichiers connexes et ainsi éviter des problèmes tel que "Qui a modifié le fichier ZaZa, tout fonctionnait hier et aujourd'hui, il y a des bugs !" Avec Git, vous retrouverez sans problème la version qui fonctionnait la veille.



- **GitHub** est une plateforme de développement inspirée par votre manière de travailler. De l'open source à l'entreprise, vous pouvez héberger et revoir du code, gérer des projets et construire des logiciels aux côtés de millions d'autres développeurs. **GitHub** réunit les équipes pour trouver des solutions ensemble, faire évoluer les idées tout en s'enrichissant mutuellement.



VI. MISE EN ŒUVRE DU PROJET

Architecture logicielle du systeme

Pour le développement, nous avons appliqué le modèle MVC.

Ce paradigme divise l'IHM (Interface Homme Machine) en un modèle (M pour modèle de données) une vue (V pour la présentation, l'interface utilisateur) et un contrôleur (C pour la logique de contrôle, et la gestion des événements / synchronisation), chacun ayant un rôle précis dans l'interface.

Ce patron d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

Le modele:

Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

La vue:

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Le controleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une

action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier notifie la vue que les données ont changée pour qu'elle les mette à jour.

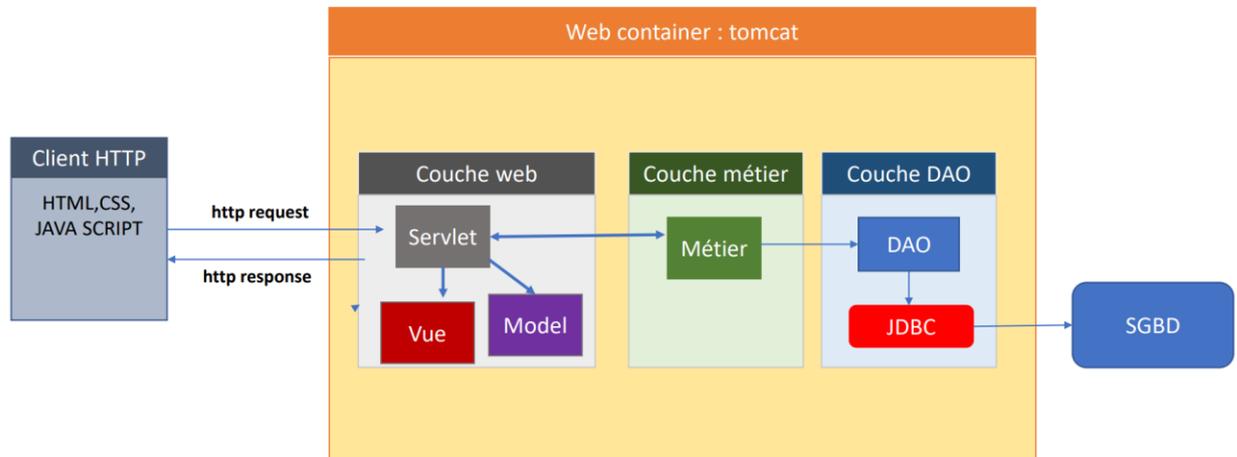


Figure 7: Architecture logicielle du projet

La principale caractéristique de cette architecture est la séparation des préoccupations (Données, service et présentation) grâce à la séparation stricte des couches applicatives. En effet, on peut observer les trois couches de l'application :

- **Couche DAO** : permet les accès à la base de données.
- **Couche métier** : contient l'ensemble du code service, elle organise les accès à la couche DAO et ses aspects transactionnelles.
- **Couche Web** : cette couche est la couche d'entrée dans l'application du point de vue d'utilisateur. Elle appelle les traitements de la couche service en fonction des actions effectuées par l'utilisateur et récupère les données retournées.

VII. DEMONSTRATION

Une vue sur la structure de notre projet : package beans qui contient les différentes classes qui seront transformées à des tables dans la base de données grâce à la technologie JPA/Hibernate.

Package DAO qui contient des interfaces qui rassemble les fonctions qui communiquent avec la base de données et leurs implémentations dans des classes. Exp : IEtudiantDAO , IEtudiantImpIDAO .



Voici un exemple d'une implémentation d'une classe dans le projet, et les annotations qui permettent à la JPA de générer automatiquement la table correspondant à cette classe dans la base de données :

```
Etudiant.java x
1 package beans;
2
3 //import java.util.Date;
4 import java.util.List;
5
6
7
8 @Entity
9 @Table(name = "Etudiant")
10 public class Etudiant {
11     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     @Column(name = "id")
14     private int id;
15
16     @Column(name = "nom")
17     private String nom;
18
19     @Column(name = "prenom")
20     private String prenom;
21
22     @Column(name = "sexe")
23     @Enumerated(EnumType.STRING)
24     private Gender sexe;
25
26     @Column(name = "cin")
27     private String cin;
28
29     @Column(name = "cne")
30     private String cne;
31
32     @Column(name = "dateInscription")
33     private Date dateInscription;
34
35     @Column(name = "dateNaissance")
36     private Date dateNaissance;
37
38     @Column(name = "email")
39     private String email;
```

Voici une vue générale sur les tables qui composent la base de données de l'application.

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length
absence	InnoDB	10	Dynamic	6	8192	48.0 KiB
anneeuniversitaire	InnoDB	10	Dynamic	2	8192	16.0 KiB
document	InnoDB	10	Dynamic	2	32768	64.0 KiB
enseignements	InnoDB	10	Dynamic	11	1489	16.0 KiB
etudiant	InnoDB	10	Dynamic	2	8192	16.0 KiB
filiere	InnoDB	10	Dynamic	3	5461	16.0 KiB
inscription_adm	InnoDB	10	Dynamic	0	0	16.0 KiB
matiere	InnoDB	10	Dynamic	9	1820	16.0 KiB
me_absence	InnoDB	10	Dynamic	0	0	16.0 KiB
module	InnoDB	10	Dynamic	7	2340	16.0 KiB
professeur	InnoDB	10	Dynamic	9	1820	16.0 KiB
reservation	InnoDB	10	Dynamic	2	8192	16.0 KiB
salles	InnoDB	10	Dynamic	2	8192	16.0 KiB
schedule	InnoDB	10	Dynamic	0	0	16.0 KiB
semestre	InnoDB	10	Dynamic	4	4096	16.0 KiB
user	InnoDB	10	Dynamic	31	528	16.0 KiB

Une description courte (Table – Column-Type) sur les différentes colonnes de chaque table de la base de données. (ces données sont prises de MySQL Workbench).

Table	Column	Type
absence	creneau	varchar(255)
absence	date	date
absence	Etudiant_Id	int
absence	id	int
absence	justificatif	blob
absence	Matiere_Id	int
absence	presence	bit(1)
anneeuniversitaire	date_debut	date
anneeuniversitaire	date_fin	date
anneeuniversitaire	id	int
document	Etudiant_id	int
document	file	blob
document	id	int
document	type	varchar(255)
enseignements	Matiere_id	int
enseignements	Professeur_id	int
etudiant	cin	varchar(255)
etudiant	cne	varchar(255)
etudiant	dateInscription	datetime(6)
etudiant	dateNaissance	datetime(6)
etudiant	email	varchar(255)
etudiant	etablissement	varchar(255)
etudiant	filiere_id	int
etudiant	id	int

etudiant	◇ lieuBac	varchar(255)
etudiant	◇ lieuNaissance	varchar(255)
etudiant	◇ lycee	varchar(255)
etudiant	◇ mention	varchar(255)
etudiant	◇ nationalite	varchar(255)
etudiant	◇ nom	varchar(255)
etudiant	◇ prenom	varchar(255)
etudiant	◇ province	varchar(255)
etudiant	◇ sexe	tinyint
etudiant	◇ User_id	int
etudiant	◇ ville	varchar(255)
filiere	◇ id	int
filiere	◇ nomFiliere	varchar(255)
inscription_adm	◇ Annee_id	int
inscription_adm	◇ Etudiant_id	int
matiere	◇ id	int
matiere	◇ module_id	int
matiere	◇ nomMatiere	varchar(255)
me_absence	◇ absence_Id	int
me_absence	◇ etudiant_Id	int
me_absence	◇ id	int
me_absence	◇ Matiere_Id	int
module	◇ id	int
module	◇ nomModule	varchar(255)
module	◇ semestre_id	int
professeur	◇ cin	varchar(255)
professeur	◇ email	varchar(255)
professeur	◇ id	int
professeur	◇ nationalite	varchar(255)
professeur	◇ nom	varchar(255)
professeur	◇ prenom	varchar(255)
professeur	◇ province	varchar(255)
professeur	◇ sexe	varchar(25)
professeur	◇ User_id	int
professeur	◇ ville	varchar(255)
reservation	◇ crenau	varchar(255)
reservation	◇ date	date
reservation	◇ Professeur_Id	int
reservation	◇ reservation_id	int
reservation	◇ Salle_Id	int
salles	◇ description	varchar(120)
salles	◇ id	int
salles	◇ numero	int
salles	◇ occupation	bit(1)
salles	◇ typesalle	varchar(255)
schedule	◇ AnneeUniversitaire_id	int
schedule	◇ id	int
schedule	◇ Semester_id	int
schedule	◇ session	varchar(255)
semestre	◇ filiere_id	int
semestre	◇ id	int
semestre	◇ nomSemestre	varchar(255)
user	◇ id	int
user	◇ password	varchar(255)
user	◇ role	varchar(255)
user	◇ username	varchar(255)

Remarque : il y a certaines variables qui ont le même nom de code, leur existence est autorisée, puisqu'ils seront dans différentes tables de la base de données.

Page de connexion :

BACHELOR MEKNES

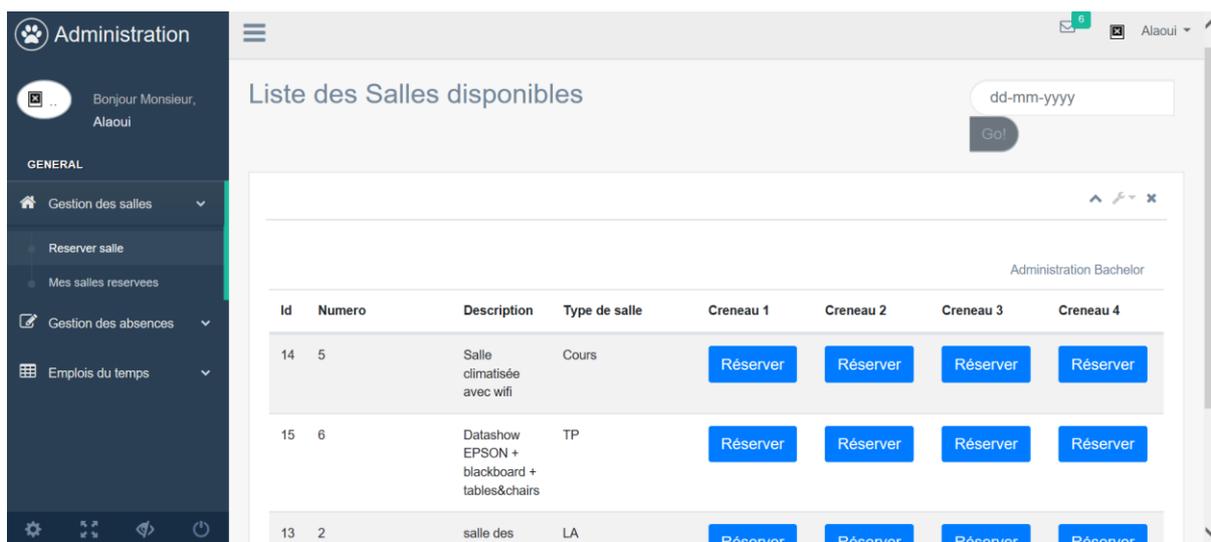
[log in](#) [Lost your password?](#)

[New to site? Create Account](#)

 **Administration Bachelor!**

©2016 All Rights Reserved. Gentelella Alela! is a Bootstrap 3 template. Privacy and Terms

Page de réservation des salles par professeur :



The screenshot shows the 'Administration Bachelor' interface. On the left is a dark sidebar menu with the following items: 'Administration', 'Bonjour Monsieur, Alaoui', 'GENERAL', 'Gestion des salles' (with a dropdown arrow), 'Reserver salle', 'Mes salles reservees', 'Gestion des absences' (with a dropdown arrow), and 'Emplois du temps' (with a dropdown arrow). At the bottom of the sidebar are icons for settings, a calendar, a speech bubble, and a power button.

The main content area is titled 'Liste des Salles disponibles' and includes a date selection field 'dd-mm-yyyy' with a 'Go!' button. Below this is a table of available rooms. The table has columns for 'Id', 'Numero', 'Description', 'Type de salle', and four 'Creneau' (slot) columns. Each cell in the 'Creneau' columns contains a blue 'Réserver' button.

Id	Numero	Description	Type de salle	Creneau 1	Creneau 2	Creneau 3	Creneau 4
14	5	Salle climatisée avec wifi	Cours	Réserver	Réserver	Réserver	Réserver
15	6	Datashow EPSON + blackboard + tables&chairs	TP	Réserver	Réserver	Réserver	Réserver
13	2	salle des	LA	Réserver	Réserver	Réserver	Réserver

Page d'ajout d'absence :

Bonjour Monsieur, Alaoui

GENERAL

- Gestion des salles
- Gestion des absences
- Emplois du temps

Ajout d'absences

Java1

dd-mm-yyyy

C1

Go!

Listes des étudiants

Show 10 entries

Search:

<input type="checkbox"/>	Id	Nom	Prenom	CIN	CNE	Email	Ville	Sexe
<input checked="" type="checkbox"/>	1	mohamed	hafidi			mohamed.hafidi@gmail.com		
<input type="checkbox"/>	2	hakmaoui	ali					

Showing 1 to 2 of 2 entries

Previous 1 Next

Ajouter Absence

Page inscription administrative des étudiants :

Administration

Bonjour, Admin

TACHES ADMINISTRATIVES

- Gestion des absences
- Gestion des étudiants
- Ajouter étudiant
- Liste des étudiants
- Génération des documents
- Gestion des salles
- Gestion des modules
- Gestion des professeurs

Inscription administrative

Search for... Go!

Formulaire

Informations personnelles de l'étudiant

Nom* Last name

Prénom* first name

CNE* Ex: M124578451

CIN* Ex: D784512

email*

Date de naissance*

Page de gestion des professeurs :

The screenshot shows the 'Liste des Professeurs' page. On the left is a dark sidebar with the 'Administration' header and a list of administrative tasks. The main content area has a title 'Liste des Professeurs' and two dropdown menus for 'S1' and 'BGI', with a 'Go!' button. Below is a table of professors with columns for Id, Nom, Prenom, CIN, Email, Ville, Nationalite, and Sexe. Each row has 'Modifier' and 'Supprimer' buttons.

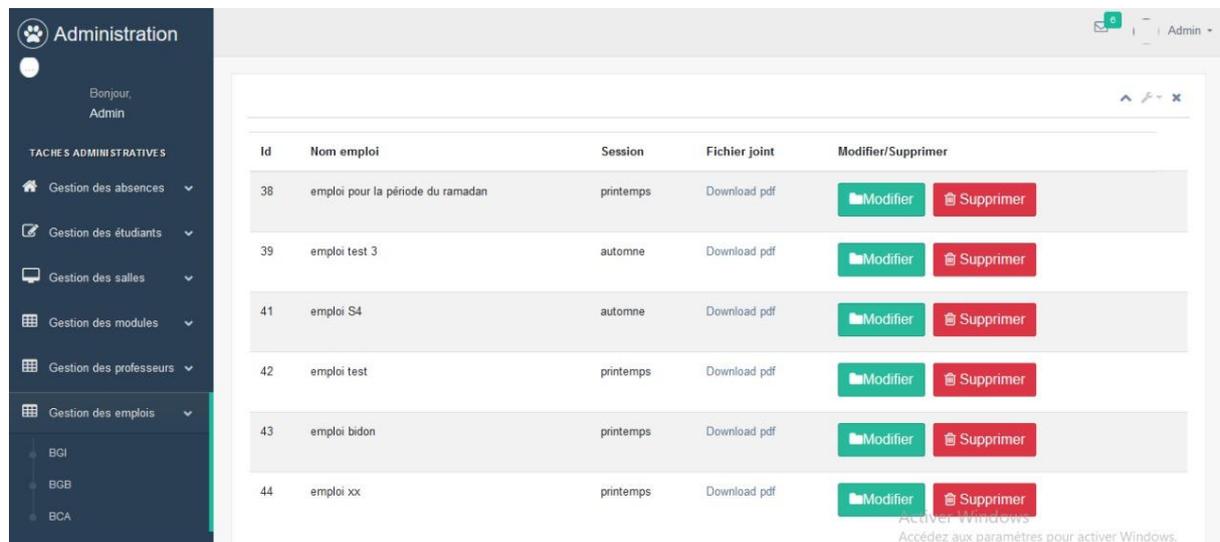
Id	Nom	Prenom	CIN	Email	Ville	Nationalite	Sexe	Modifier/Supprimer
1	Bekri	Ali	VA1234	ali.bekri@yahoo.com	Midelt	marocaine		Modifier Supprimer
2	Hafidi	Mohamed	VA23641	mohamed.hafidi.mh@gmail.com	Meknes	marocaine		Modifier Supprimer

Page de gestion des étudiants :

The screenshot shows the 'Ajouter étudiant' page. The sidebar is similar to the previous page, but 'Gestion des étudiants' is highlighted. The main area contains a form with fields for 'Date de début de l'année *', 'Date de fin de l'année *', 'Filière', and 'Semestre'. A green 'Chercher dans la liste des étudiants' button is below the form. At the bottom, a table shows a list of students with columns for Id, CIN, Nom, Prénom, CNE, and email. Each row has 'Modifier' and 'Supprimer' buttons.

Id	CIN	Nom	Prénom	CNE	email	Modifier/Supprimer
199	D1111	MAHOUBI	asmae	M2222222	eee@gmail.com	Modifier Supprimer

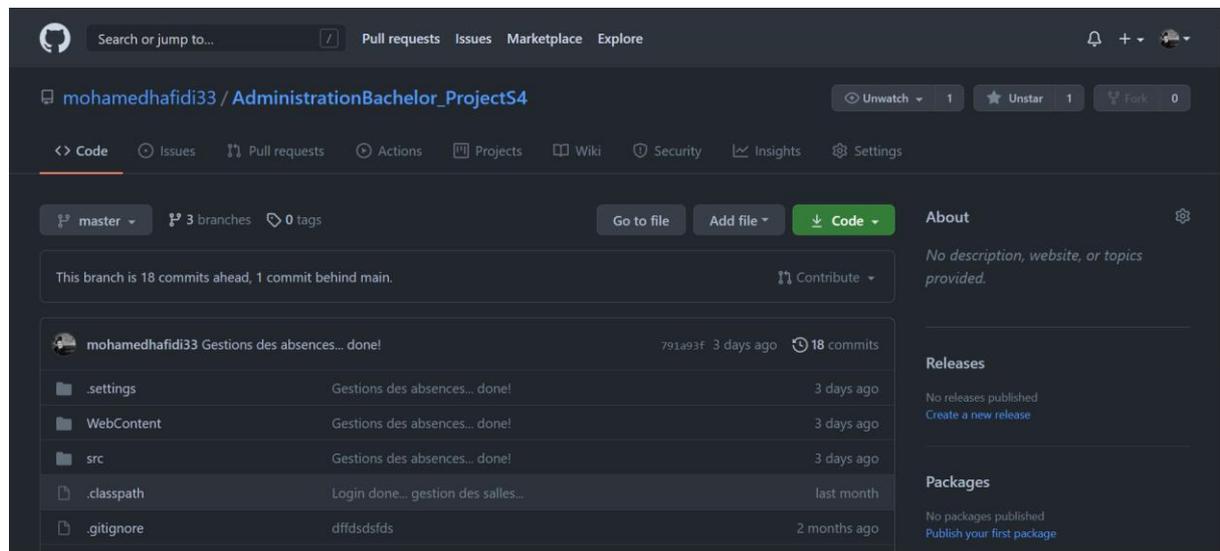
Gestion des emplois du temps :



The screenshot shows a web application interface for managing job schedules. On the left is a dark sidebar with the title 'Administration' and a user profile for 'Admin'. Below the profile is a list of administrative tasks: 'Gestion des absences', 'Gestion des étudiants', 'Gestion des salles', 'Gestion des modules', 'Gestion des professeurs', and 'Gestion des emplois'. The 'Gestion des emplois' menu item is highlighted. The main content area displays a table with the following data:

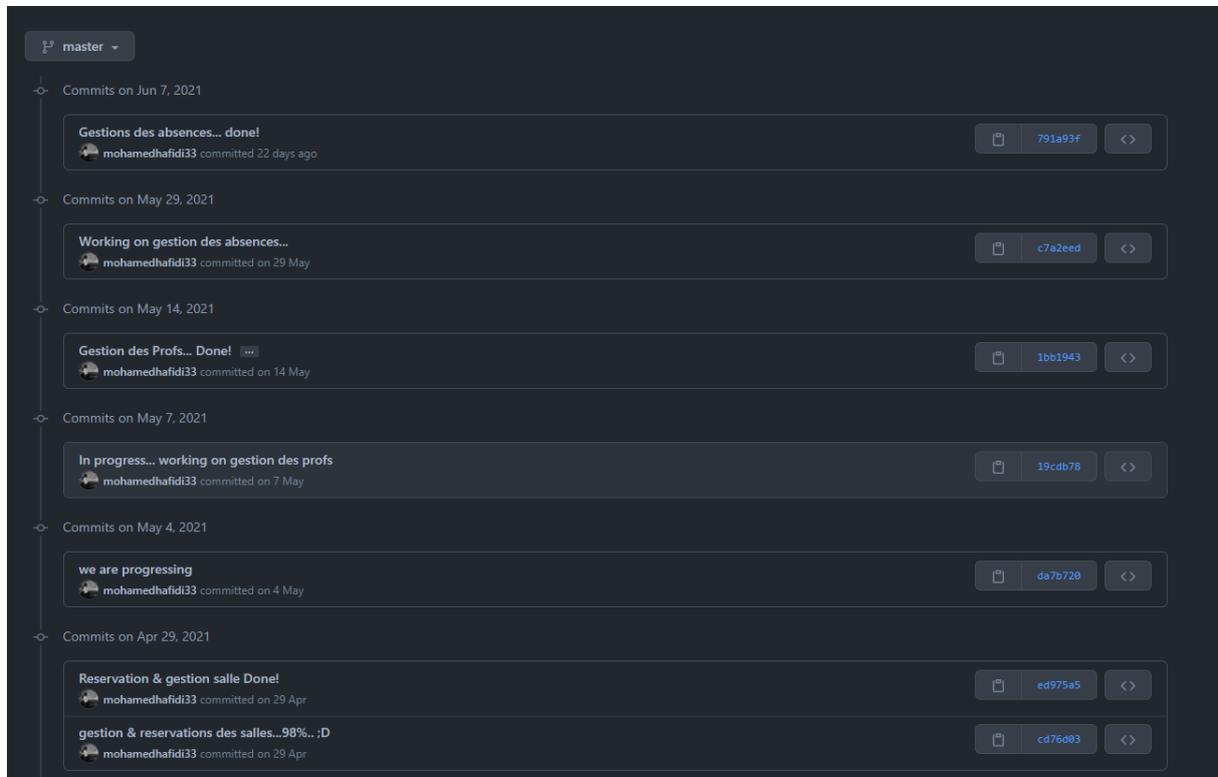
Id	Nom emploi	Session	Fichier joint	Modifier/Supprimer
38	emploi pour la période du ramadan	printemps	Download pdf	Modifier Supprimer
39	emploi test 3	automne	Download pdf	Modifier Supprimer
41	emploi S4	automne	Download pdf	Modifier Supprimer
42	emploi test	printemps	Download pdf	Modifier Supprimer
43	emploi bidon	printemps	Download pdf	Modifier Supprimer
44	emploi xx	printemps	Download pdf	Modifier Supprimer

Une vue sur notre projet sur Github :

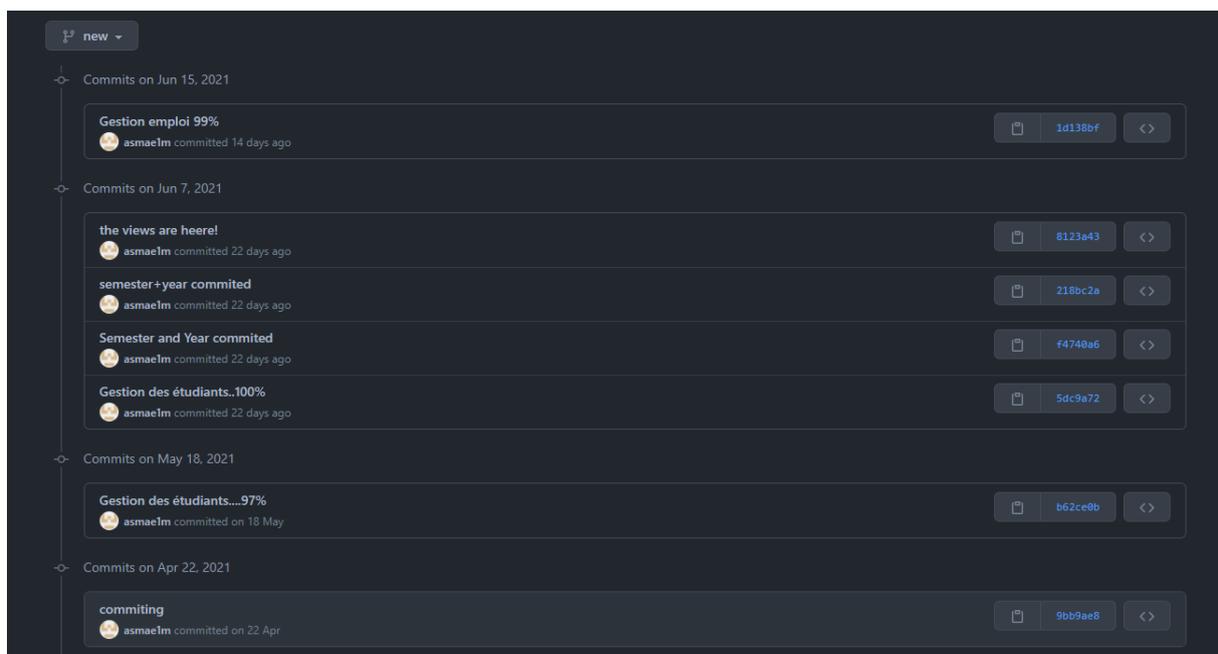


The screenshot shows a GitHub repository page for 'mohamedhafidi33 / AdministrationBachelor_ProjectS4'. The repository is on the 'master' branch, which is 18 commits ahead of the main branch. The repository contains several files and folders, including .settings, WebContent, src, .classpath, and .gitignore. The repository has 1 star and 0 forks. The page also shows sections for 'About', 'Releases', and 'Packages', all of which are currently empty.

Voici deux captures prises de notre projet sur Github qui montrent le progrès de développement de notre projet avec les dates (Commits & Pushes) .



Branch : Master (Mohamed Hafidi)



Branch : new (Asmae Mahjoubi)

VIII. CONCLUSION ET PERSPECTIVES

En conclusion, nous devons avouer que nous avons fait de notre mieux pour aboutir à ce résultat. Certes si les circonstances nous le permettaient le résultat sera encore plus développé et plus satisfaisant. Au sein du projet nous avons trouvé des difficultés comme, en premier lieu, nous voulions continuer à développer notre application partie DAO avec Hibernate/JPA, mais nous avons des problèmes que nous essayions de résoudre .

En deuxième lieu, afin de travailler ensemble sur le même projet n'était pas facile, c'est pour cela qu'on est dirigé à la technologie Git/Github, que nous avons déjà travaillé ensemble avec dans le dernier semestre.

Toutes les fonctionnalités prévues par le cahier des charges ont été réalisées : de la gestion des étudiants, professeurs et salles à l'emploi du temps.

Comme perspectives éventuelles d'amélioration de cette application, il est possible d'ajouter les fonctionnalités suivantes :

- Changer la description des salles (possède un réseau local, wifi, projecteur, tableau blanc ...), de telle sorte que l'on peut cocher des carreaux, au lieu de donner une description textuelle de ce qu'une salle possède à l'intérieur.
- Alimenter la table "absence" par importation Excel possédant les absences par classe.

La programmation d'application web est une tendance de plus en plus adoptée par les entreprises en raison de ses nombreux avantages. Ce qui rend ce travail très utile pour notre formation, car il nous a permis de découvrir certains mécanismes de développement Web, et aussi d'améliorer nos connaissances au niveau du langage Java, en espérant d'entamer des autres technologies de développement Web dans le futur.

IX. BIBLIOGRAPHIE

- ✚ Site officiel d'Apache tomcat: <https://tomcat.apache.org/>
- ✚ Site officiel de Github : <https://github.com/>
- ✚ Site officiel de Maven : <https://maven.apache.org/>
- ✚ Site officiel de Bootstrap : getbootstrap.com/
- ✚ Site officiel de Hibernate : hibernate.org/
- ✚ Projet sur Github :
github.com/mohamedhafidi33/AdministrationBachelor_ProjectS4
- ✚ Documentation Officiel d'Eclipse: <https://help.eclipse.org/2020-06/index.jsp>