République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques Informatique
Filière : Informatique
Spécialité : Systèmes d'informatique
N° de l'Arrêté d'habilitation de la spécialité : arrêté n °872 du 26/07/2016

Mémoire de fin d'études en vu de l'obtention du Diplôme de Licence Académique

Thème

Développement d'un Système de Planification des Tests et Séances de Rattrapage

Présenté par :
BENHADDADI Lina
BOUGUERRA Rayan
GUETTARA Chaima
SIAR Nour El Houda

Stage Pratique réalisé au : /

Soutenu le 04/06/2026 Devant le jury composé de
M .DJERBI Rachid : Examinateur
M.DJOUZI Kheyreddine : Encadreur

Résumé

Dans le cadre de la transition numérique des établissements universitaires, la planification manuelle des tests et des séances de rattrapage constitue une tâche complexe et chronophage en raison des contraintes liées aux horaires, aux salles et à la coordination des enseignants. Ce projet propose ainsi le développement d'un système de planification dédié.

La plateforme vise à faciliter la gestion pour les administrateurs, à permettre aux enseignants de soumettre et consulter les demandes, et à offrir aux étudiants un accès simplifié aux sessions planifiées. Un système de notifications assure également une meilleure communication entre les acteurs.

Ce travail a permis de mobiliser des compétences en développement logiciel, bases de données et conception des systèmes d'information.

Enfin, ce projet contribue à améliorer l'efficacité et la précision de la gestion académique au sein des établissements d'enseignement supérieur.

Mots-clés : Planification, Tests, Séances de rattrapage, Gestion académique.

Abstract

In the context of the digital transformation of higher education institutions, manual planning of tests and retake sessions has become a complex and time-consuming task due to constraints related to schedules, room availability, and teacher coordination. This project therefore proposes the development of a dedicated planning system.

The platform aims to facilitate management for administrators, allow teachers to submit and consult requests, and provide students with simplified access to scheduled sessions. A notification system also ensures better communication between stakeholders.

This work has made it possible to apply skills in software development, databases, and information system design.

Finally, this project contributes to improving the efficiency and accuracy of academic management within higher education institutions.

Keywords : Planning, Examinations, Make-up Sessions, Academic Management

Avant-propos

Nous remercions avant tout ***ALLAH***, le Tout-Puissant, de nous avoir accordé la santé, la volonté et la persévérance nécessaires pour mener à bien ce travail.

La réalisation de ce projet de fin d'études représente une étape importante dans notre parcours universitaire. Ce travail nous a permis de mettre en pratique les connaissances acquises durant nos années d'étude dans le domaine du développement informatique, tout en développant notre esprit d'analyse, de conception et de recherche.

Nous espérons que cette application pourra apporter une contribution utile à l'amélioration de la gestion universitaire et ouvrir la voie à de futures perspectives d'amélioration et d'évolution.

Nous exprimons notre profonde gratitude à notre encadrant, ***M. DJOUZI***, pour son accompagnement précieux, sa disponibilité, ses orientations pertinentes ainsi que ses conseils avisés tout au long de la réalisation de ce projet. Son soutien scientifique et ses encouragements ont été d'une grande importance dans l'aboutissement de ce mémoire.

Nous adressons également nos sincères remerciements aux honorables membres du jury pour l'intérêt qu'ils portent à notre travail en acceptant de l'examiner et de l'évaluer.

Nos remerciements s'étendent à l'ensemble des enseignants du département d'informatique pour la qualité de la formation dispensée, ainsi que pour les connaissances et compétences acquises durant notre parcours universitaire.

Nous tenons aussi à exprimer notre reconnaissance envers toutes les personnes ayant contribué, de près ou de loin, à la réalisation de ce travail.

Enfin, nous adressons une pensée particulière à nos familles et à nos proches pour leur soutien moral, leur patience et leurs encouragements permanents tout au long de notre cursus universitaire.

DÉDICACE

Nous dédions ce modeste travail, avec un profond amour et une grande reconnaissance :

À nos mères, qui nous ont toujours entourés de tendresse et d'espoir, sources d'amour inépuisable, et qui nous ont soutenus par leurs prières.

À nos pères, nos soutiens dans la vie, qui nous ont guidés et encouragés à avancer et à réussir.

À nos sœurs, nos frères et à toute notre grande famille, pour leur amour et leur soutien.

À tous nos amis, avec qui nous avons partagé de merveilleux moments de joie et de bonheur.

Nous remercions également notre encadreur **M. DJOUZI**, *pour ses précieux conseils et son soutien tout au long de ce travail.*

Chaima, Lina, Nour El Houda, Rayane

Table des matières

Liste des abréviations **ix**

Introduction Générale **1**

I Contexte Général et État de l'Art **2**

I.1 Introduction 3

I.2 Contexte général 3

I.2.A Environnement académique 3

I.2.B Gestion des tests 3

I.2.C Gestion des sèances de rattrapage 4

I.2.D Défis Actuels 4

I.3 État de l'art 5

I.4 Problématique 5

I.5 Suggestions des solutions futures 6

I.6 Conclusion 6

II Analyse et conception **7**

II.1 Introduction 8

II.2 Modélisation 8

II.2.A Définition de la modélisation 8

II.2.B Intérêt de la modélisation 8

II.2.C Présentation de quelques méthodes de modélisation 9

II.2.C.a UML 9

II.2.C.b MERISE 9

II.2.D Étude comparative de ces méthodes 10
II.2.E Démarche de modélisation adoptée 10
II.3 Analyse 11
II.3.A Identification des acteurs 11
II.3.B Spécification des besoin 12
II.3.B.a Besoins fonctionnels 12
II.3.B.b Besoins non fonctionnels 13
II.3.C Spécification des tâches 13
II.3.D Diagramme de cas d'utilisation 15
II.4 Conception 16
II.4.A Diagramme de séquence (DS) 16
II.4.A.a DS pour le cas d'utilisation "S'authentifier" en tant qu'étudiant 16
II.4.A.b DS pour le cas "Consulter les créneaux des tests" en tant qu'admin 17
II.4.A.c DS pour le cas "Proposer une séance de rattrapage" en tant qu'enseignant 17
II.4.B Diagramme d'activités 18
II.4.B.a Diagramme d'activités pour le cas d'utilisation "Annuler une séance de test" en tant qu'enseignant 18
II.4.B.b Diagramme d'activités pour le cas d'utilisation "Envoyer les notifications des demandes traitées" en tant qu'enseignant 19
II.4.C Dictionnaire des données épuré 19
II.4.D Règles de gestion 20
II.4.E Diagramme de classe 22
II.4.F Diagramme de paquetage 22
II.4.G Diagramme de déploiement 23
II.4.H Modèle Logique de Données 24
II.4.I Modèle Physique de Données 24
II.5 Conclusion 25

III Réalisation **26**

III.1 Introduction 27
III.2 Présentation de l'environnement logiciel utilisé 27
III.2.A Outils et logiciels de conception et de développement utilisés 27
III.2.A.a Draw.io 27
III.2.A.b Figma 27
III.2.A.c Visual Studio Code 28
III.2.A.d XAMPP 28
III.2.A.e Composer 28
III.2.B Langages de programmation et frameworks utilisés 28
III.2.B.a HTML 28
III.2.B.b CSS 28
III.2.B.c Java Script 29
III.2.B.d PHP 29
III.2.B.e SQL 29
III.2.B.f Bootstrap CSS 29
III.2.B.g Laravel 29
III.3 Présentation de l'environnement matériel utilisé 29
III.4 Conclusion 30

Conclusion Générale **31**

Table des figures

II.1 Démarche UML à adopter 11
II.2 Diagramme de contexte de notre système 12
II.3 Diagramme de cas d'utilisation détaillé de notre système 15
II.4 Diagramme de séquence pour le cas "S'authentifier" en tant qu'étudiant. . 16
II.5 DS pour le cas "Consulter les créneaux des tests" en tant qu'administrateur 17
II.6 DS pour le cas "Proposer une séance de rattrapage" en tant qu'enseignant 17
II.7 Diagramme d'activités pour le cas d'utilisation "Annuler une séance de test" en tant qu'enseignant 18
II.8 Diagramme d'activités pour le cas d'utilisation "Envoyer les notifications des demandes traitées" en tant qu'enseignant 19
II.9 Diagramme de classe de notre système 22
II.10 Diagramme de paquetage de notre système 23
II.11 Diagramme de déploiement de notre système. 23
II.12 Modèle physique de données 25

Liste des tableaux

I.1 Table comparative des plateformes de gestion du temps et planification. . . 5

II.1 Table comparative entre MERISE et UML 10
II.2 Identification des acteurs et description de leurs tâches 13
II.3 Dictionnaire du données épuré de notre système 20

Liste des abréviations

BDD	Base Des Données
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DS	Diagramme de Séquence
GHz	GigaHertz
Go	Giga-octet
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
MCD	Modèle Conceptuel de Données
MCT	Modèle Conceptuel de Traitements
MERISE	Méthode d'Étude et de RéalisationInformatique pour les Systèmes d'Entreprise
MLD	Modèle Logique de Données
MOT	Modèle Organisationnel de Traitements
MPD	Modèle Physique de Données
OMG	Object Management Group
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
OS	Operating System
PHP	Hypertext Preprocessor
RAM	RandomAccess Memory
SGBD	Système de Gestion de Bases de Données
SQL	Structured Query Language
UML	Unified Modeling Language

Introduction Générale

Avec le développement des technologies de l'information et de la communication, la numérisation des services administratifs devient essentielle dans l'enseignement supérieur. Les universités modernisent leurs systèmes de gestion afin d'améliorer l'organisation pédagogique, réduire les erreurs et faciliter la communication.

Cependant, la planification des tests et des rattrapages reste une tâche complexe nécessitant une bonne coordination entre les acteurs académiques. Dans plusieurs établissements, cette gestion manuelle entraîne des problèmes tels que les conflits d'horaires, une mauvaise allocation des salles et une perte de temps importante.

Dans ce contexte, le développement d'un système informatique de planification et de gestion des tests et des séances de rattrapage constitue une solution efficace permettant d'automatiser ces opérations. Ce système offre la possibilité de centraliser les données, faciliter l'organisation des examens et améliorer la communication entre les utilisateurs.

L'objectif de ce projet est de concevoir et développer une application capable de gérer efficacement la programmation des tests et des séances de rattrapage tout en respectant les contraintes pédagogiques et organisationnelles.

Ce mémoire est organisé en trois (03) chapitres :

- **Chapitre 1 :** Contexte général du projet Ce chapitre présente le cadre général du projet, les problématiques rencontrées, les objectifs visés ainsi qu'une étude des solutions existantes.
- **Chapitre 2 :** Analyse et conception du système Ce chapitre est consacré à l'analyse des besoins fonctionnels et non fonctionnels, ainsi qu'à la conception de l'application à travers les différents diagrammes et modèles utilisés.
- **Chapitre 3 :** Réalisation et implémentation de l'application Ce chapitre décrit les outils et technologies adoptés, les différentes étapes de développement de l'application, ainsi que les interfaces et les fonctionnalités réalisées.

Enfin, le manuscrit se termine par une conclusion générale et une discussion sur les perspectives futures de la solution proposée.

Chapitre I

Contexte Général et État de l'Art

Résumé : *Ce chapitre présente le contexte général du projet ainsi que la problématique liée à l'organisation des examens et des séances de rattrapage. La méthode actuelle repose sur un travail manuel, ce qui engendre plusieurs problèmes tels que les conflits d'horaires, la difficulté de gestion des salles et une communication insuffisante avec les étudiants. Pour remédier à ces problèmes, plusieurs solutions ont été proposées, notamment le développement d'un système informatique permettant une meilleure organisation, une réduction des erreurs et une amélioration de la gestion des ressources ainsi que de la communication.*

Sommaire

I.1 Introduction **3**
I.2 Contexte général **3**
I.3 État de l'art **5**
I.4 Problématique **5**
I.5 Suggestions des solutions futures **6**
I.6 Conclusion **6**

I.1 Introduction

L'organisation des tests et des séances de rattrapage est essentielle dans les établissements universitaires. Avec l'augmentation du nombre d'étudiants, leur planification devient plus complexe. L'utilisation d'un système informatique permet d'améliorer leur gestion et leur organisation.Ce chapitre présente le contexte général du projet et la problématique.

I.2 Contexte général

Ce contexte académique nécessite une gestion rigoureuse des informations liées aux étudiants, aux enseignants, aux modules et aux examens afin d'assurer le bon déroulement des activités pédagogiques. Dans cette perspective, la mise en place d'un système de gestion informatisé permettrait d'optimiser l'organisation administrative, de réduire les erreurs et d'améliorer la communication au sein du département.

I.2.A Environnement académique

Le département d'informatique de l'Université Mohamed Bougara – Boumerdès accueille un grand nombre d'étudiants répartis sur différents niveaux d'études, allant de la Licence au Master. Il assure des formations théoriques et pratiques à travers des modules organisés par semestre, comprenant des cours magistraux, des travaux dirigés et des travaux pratiques. L'organisation académique repose sur un emploi du temps défini chaque semestre, avec des évaluations régulières sous forme de tests et d'examens. Le département dispose de ressources pédagogiques variées, telles que des laboratoires informatiques, des salles équipées et des plateformes numériques pour faciliter l'apprentissage. Cependant, certaines difficultés persistent dans la coordination des emplois du temps et la gestion des examens, ce qui justifie le développement d'un système de gestion pour améliorer l'efficacité administrative et le suivi pédagogique.

I.2.B Gestion des tests

La gestion des tests se fait actuellement de manière traditionnelle au sein du département d'informatique de l'Université M'hamed Bougara – Boumerdès. L'enseignant se rend auprès du service de planification pour demander les salles libres à la date et l'heure souhaitées, en s'assurant de la disponibilité des groupes. Ensuite, les étudiants sont informés des horaires par des affichages ou par email. Cette méthode manuelle de coordination des horaires et des salles peut parfois entraîner des conflits et rendre l'accès à l'information difficile pour certains étudiants, tout en consommant beaucoup de temps et d'efforts pour les enseignants et l'administration.

I.2.C Gestion des sèances de rattrapage

Actuellement, pour organiser les séances de rattrapage, l'enseignant va chez le responsable des plannings pour lui communiquer le jour et l'heure souhaités. Le responsable vérifie ensuite les salles disponibles pour ces horaires et confirme la réservation. Après cela, les étudiants sont informés des dates et heures des séances via des annonces ou d'autres moyens de communication. Cette méthode manuelle permet d'organiser les séances mais reste consommatrice de temps et peut entraîner des conflits de disponibilité.

I.2.D Défis Actuels

Le département d'informatique rencontre plusieurs défis liés à l'organisation des examens et des séances de rattrapage, ce qui peut affecter l'efficacité administrative et la qualité du suivi pédagogique :

- **Tâches manuelles et chronophages :** La planification des examens et des séances de rattrapage repose principalement sur des méthodes traditionnelles, comme l'utilisation de documents papier ou de fichiers non centralisés. Cette approche demande un temps considérable pour la collecte des informations, la vérification des disponibilités des enseignants et des salles, ainsi que pour la mise à jour des horaires. De plus, toute modification nécessite souvent de refaire plusieurs étapes manuellement, ce qui augmente la charge de travail pour l'administration.

- **Risque de conflits d'horaires :** L'absence d'un système informatisé centralisé peut entraîner des chevauchements entre les examens, les séances de rattrapage et les autres activités pédagogiques. Par exemple, une même salle peut être attribuée à deux activités différentes au même moment, ou un enseignant peut être programmé dans deux séances simultanément. Ces situations provoquent des perturbations dans l'organisation et nécessitent des ajustements de dernière minute.

- **Communication limitée avec les étudiants :** L'absence d'un système informatisé centralisé favorise l'apparition de conflits de planification entre les examens, les séances de rattrapage et les autres activités pédagogiques. Cette situation peut entraîner l'attribution simultanée d'une même salle à plusieurs activités ou la programmation d'un enseignant dans des séances au même horaire. Ces chevauchements perturbent l'organisation pédagogique et obligent souvent l'administration à effectuer des ajustements de dernière minute.

- **Suivi difficile de la présence :** Le suivi de la présence des étudiants pendant les séances de rattrapage se fait généralement de manière manuelle, à l'aide de listes papier ou de feuilles de présence. Cette méthode peut entraîner des erreurs, des pertes d'informations ou des difficultés dans la consultation des données ultérieurement.

De plus, l'administration ne dispose pas toujours d'une vue claire et rapide sur la participation des étudiants.

- **Manque de traçabilité et d'archivage des données :** L'utilisation de méthodes manuelles rend également difficile la conservation et la consultation des données liées aux examens et aux séances de rattrapage. Les informations peuvent être dispersées dans plusieurs documents ou fichiers, ce qui complique leur recherche et leur exploitation ultérieure pour des besoins administratifs ou statistiques.

I.3 État de l'art

La table I.1 présente quelques plateformes existantes concernant la planification et la gestion des emplois du temps.

Plateforme	**Caractéristiques principales**	**Avantages**	**Inconvénients**
Infosilem Exam [1]	Création automatisée de calendrier, planification des examens, modification partielle, attribution des salles	Création rapide, facile à modifier	Problèmes pour les étudiants, ne couvre pas tout l'emploi du temps, payant
CELCAT Timetabler [2]	Planification cours/examens, IA pour conflits, interface drag-drop, publication temps réel	Très flexible, contrôle total, réduction des conflits	Coût élevé, formation nécessaire
UniTime [3]	Planification automatique, gestion des contraintes, attribution des salles	Open source, réduit les erreurs, gain de temps	Mise en place complexe,Besoin d'une équipe technique
aSc Timetables [4]	Planification cours/examens, gestion enseignants et salles	Simple d'utilisation, adapté écoles/lycées	Moins performant pour universités
Units [5]	Gestion complète, application mobile, gestion absences	Communication facilitée, suivi efficace	Payant, configuration longue

TABLE I.1 – Table comparative des plateformes de gestion du temps et planification.

I.4 Problématique

Le département d'informatique rencontre plusieurs difficultés dans l'organisation des tests et des séances de rattrapage, en raison de l'utilisation de méthodes traditionnelles et d'une coordination manuelle avec le responsable des emplois du temps. Les principaux problèmes peuvent être résumés comme suit :

- **Organisation manuelle des séances :** La planification des tests et des séances de rattrapage dépend d'une communication directe entre les enseignants et le responsable des emplois du temps, ce qui rend le processus lent et parfois inefficace.
- **Risque de conflits d'horaires :** L'absence d'un système automatisé peut entraîner des chevauchements entre les séances de rattrapage, les tests et les autres activités pédagogiques.

- **Gestion complexe des salles :** La recherche de salles disponibles pour organiser les tests ou les séances de rattrapage peut prendre beaucoup de temps, surtout lorsqu'il faut vérifier la capacité des salles et leur disponibilité.
- **Accès limité à l'information pour les étudiants :** Les étudiants ne disposent pas toujours d'un moyen rapide et clair pour consulter les horaires des tests et des séances de rattrapage.
- **Charge administrative importante :** Les enseignants et l'administration doivent consacrer beaucoup de temps à l'organisation, à la vérification des horaires et à la communication des informations.
- **Manque de centralisation des données :** Les informations concernant les séances, les salles et les groupes peuvent être dispersées dans plusieurs documents, ce qui complique leur gestion et leur mise à jour.

I.5 Suggestions des solutions futures

Pour résoudre les problèmes liés à l'organisation des tests et des séances de rattrapage, les solutions suivantes peuvent être proposées :

- **Amélioration de la méthode traditionnelle :** Optimiser la coordination manuelle en utilisant des tableaux partagés simples, afin de réduire les conflits d'horaires et garantir l'accès aux informations pour les étudiants.
- **Développement d'un système informatique complet :** Concevoir un logiciel permettant de planifier automatiquement les tests et les séances de rattrapage, de gérer les salles, les groupes et les horaires, tout en offrant un accès rapide et précis aux informations pour les étudiants et les enseignants.
- **Utilisation d'applications externes existantes :** Recourir temporairement à des outils comme Google Calendar ou Calendly pour organiser les horaires et suivre les activités, en attendant le développement du système spécifique du département.

La deuxième solution est considérée comme la plus efficace, car elle réduit les erreurs, économise du temps et facilite l'organisation des tests et des séances de rattrapage de manière plus fluide et précise.

I.6 Conclusion

Dans ce chapitre, nous avons présenté le contexte général de notre projet ainsi que les principales difficultés liées à la gestion des tests et des séances de rattrapage, ce qui nous a permis de définir la problématique et de proposer des solutions. Nous allons présenter l'analyse et la conception du système proposé dans le chapitre suivant.

Chapitre II

Analyse et conception

Résumé : *Dans ce chapitre, la partie analyse et conception du système sera présentée. La modélisation ainsi que les différentes méthodes de modélisation seront abordées, afin de mettre en évidence qu'UML constitue la méthode la plus utilisée. Les éléments principaux du système, notamment les acteurs et les différentes tâches, seront ensuite définis. Enfin, la conception du système sera réalisée à l'aide des diagrammes UML, en particulier les six diagrammes UML, ainsi que les modèles logique et physique des données.*

Sommaire

II.1 Introduction **8**
II.2 Modélisation **8**
II.3 Analyse **11**
II.4 Conception **16**
II.5 Conclusion **25**

II.1 Introduction

L'analyse et la conception sont des étapes fondamentales dans le développement d'un site web. L'analyse nous permet de déterminer les besoins en décomposant un sujet complexe en éléments plus petits afin de faciliter sa compréhension. La conception, quant à elle, nous fournit une solution théorique ou une structure technique répondant aux exigences, mais cela ne signifie pas pour autant une mise en œuvre immédiate. Dans ce chapitre, nous allons essayer d'étudier le processus de modélisation, l'analyse des exigences et la conception technique en suivant une approche structurée.

II.2 Modélisation

La modélisation représente une étape essentielle dans la conception des systèmes informatiques. Cette section présentera sa définition, son intérêt ainsi que quelques méthodes de modélisation.

II.2.A Définition de la modélisation

Pour programmer une application (développer un logiciel), il ne convient pas de se lancer tête baissée dans l'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation son produit est un modèle. Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer. Il permet :

- De visualiser le système comme il est ou comme il devrait l'être.
- De valider le modèle vis à vis des clients.
- De spécifier les structures de données et le comportement du système.
- De fournir un guide pour la construction du système.
- De documenter le système et les décisions prises [6].

II.2.B Intérêt de la modélisation

La modélisation présente plusieurs avantages dans la conception et la gestion des systèmes d'information, parmi lesquels :

- Réaliser des analyses d'impact, une situation à laquelle nous sommes souvent confrontés dans le cadre de nos missions : lorsqu'un composant devient obsolète, qu'il doit

être mis à niveau ou migré, ou qu'un processus métier doit être modifié, nous identifions les composants concernés dans tous les domaines : fonctionnel, infrastructure, support, commercial.

– Faciliter le décommissionnement des composants , en maîtrisant tous les aspects du service ou de la fonctionnalité. Rendre le système d'information plus cohérent et plus "épuré" est devenu un objectif de plus en plus partagé

– Réfléchir pour transformer, Faire de la prospective et analyser l'écart entre un état existant et une cible

II.2.C Présentation de quelques méthodes de modélisation

La modélisation peut être réalisée à travers différentes méthodes, parmi lesquelles UML et MERISE figurent parmi les plus utilisées dans le domaine de l'analyse et de la conception des systèmes d'information.

II.2.C.a UML

Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

L'UML est une synthèse de langages de modélisation objet antérieurs : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997 puis adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017) [7].

II.2.C.b MERISE

MERISE organise la conception d'un système d'information en trois niveaux : le niveau conceptuel (quoi) qui décrit les données et les traitements, le niveau logique (qui, où, quand) qui précise leur organisation, et le niveau physique (comment) qui correspond à leur implémentation technique.

Dans les années 1970, la méthode MERISE a été créée en France afin d'organiser la conception des systèmes d'information, influencés par le modèle relationnel de Edgar Frank Codd. Elle s'est ensuite développée dans les années 1980, où elle a été largement adoptée en France et dans les pays francophones, avant d'évoluer avec le temps malgré la concurrence d'autres méthodes [8].

II.2.D Étude comparative de ces méthodes

Afin de mieux comprendre ces deux (02) méthodes de modélisation, leurs principales caractéristiques et différences sont présentées dans le tableau II.1.

Critères	MERISE	UML
Nature	méthode d'analyse et de conception de système d'information	langage de représentation d'un système d'information
Approche	Approche systémique et descendante (du général au particulier)	Approche orientée objet
Données et Traitements	Séparation stricte des données et des traitements	Association/Encapsulation des données et des traitements dans des objets
Niveaux d'abstraction	3 niveaux : Conceptuel, Logique, Physique	Pas de niveaux fixes ; utilise différentes vues (diagrammes) complémentaires
Modèles principaux	dépend de : MCD, MLD, MCT, MOT	dépend de : diagrammes de Cas d'utilisation, Classes, Séquences, Activités, etc (14 types)
Domaines d'application	Projets de gestion traditionnels et bases de données relationnelles	Développement logiciel complexe et moderne (web, mobile)
Cardinalités	Notées près de l'entité de départ (ex : 1,n)	Inversées graphiquement, notées près de la classe d'arrivée

TABLE II.1 – Table comparative entre MERISE et UML

II.2.E Démarche de modélisation adoptée

Au vu des méthodes de modélisation comparées, on peut considérer que l'UML figure parmi les plus utilisées, car c'est un langage visuel international standard, qui facilite le développement du système et permet de maîtriser la complexité lors de l'élaboration de solutions orientées objet. De plus, il facilite également la documentation et la maintenance futures du programme, et se distingue par sa capacité à s'étendre sans en modifier l'essence.

Pour bien comprendre cette méthodologie, la figure II.1 présente les différentes étapes de la modélisation, depuis l'analyse des besoins jusqu'à la conception détaillée du système.

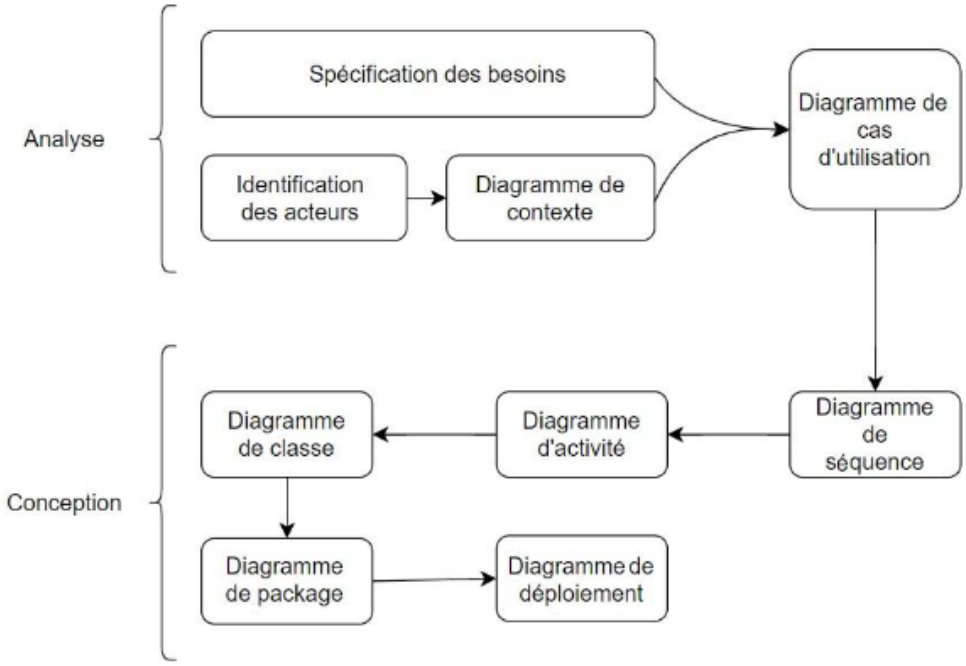


FIGURE II.1 – Démarche UML à adopter

II.3 Analyse

Dans cette partie, nous cherchons à cerner l'objectif du système en identifiant et en comprenant clairement ses exigences, tout en déterminant l'ensemble de ses contraintes, ainsi qu'en identifiant les acteurs du système et leur champ d'action.

II.3.A Identification des acteurs

Pour atteindre l'objectif du système, il est indispensable d'identifier ses principaux acteurs. C'est pourquoi nous avons déterminé les futurs utilisateurs de l'application :

- **Administrateur :** C'est l'acteur principal, chargé notamment de la planification des tests et des séances de rattrapage, de la gestion des profils des étudiants et des enseignants, ainsi que de la validation des différentes demandes.
- **Enseignant :** C'est un acteur qui peut créer un compte pour faire des demandes d'organisation des séances de rattrapage et des tests, ainsi que la consultation des plannings.
- **Étudiant :** C'est un acteur qui peut créer un compte pour consulter ses séances de rattrapage et séances de tests de son groupe.

Pour avoir une vision globale des interactions entre les acteurs et le système, nous utilisons le diagramme de contexte présenté dans la figure II.2.

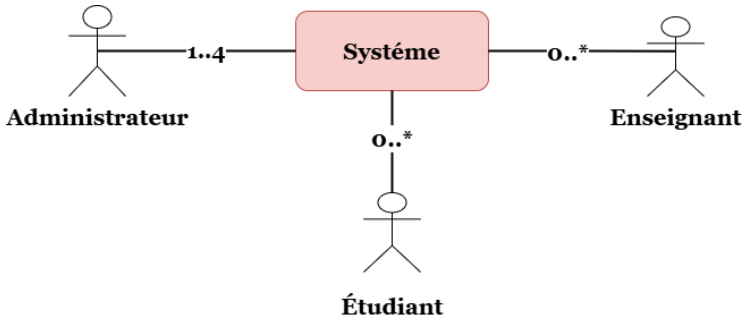


FIGURE II.2 – Diagramme de contexte de notre système

II.3.B Spécification des besoin

Dans cette section, nous allons présenter les différents besoins fonctionnels et non fonctionnels de notre système.

II.3.B.a Besoins fonctionnels

1. **Gestion des utilisateurs :** Créer, modifier et supprimer des comptes selon le rôle de chaque utilisateur (administrateur, enseignant, étudiant).

2. **Gestion des modules :** Lier chaque test ou séance de rattrapage à la matière correspondante.

3. **Gestion des salles :** Enregistrer les salles, définir la capacité et éviter les conflits d'occupation.

4. **Planification des tests :** Programmer la date, l'heure, la salle, le module et l'enseignant sans conflit.

5. **Planification des séances de rattrapage :** Organiser les séances pour les étudiants consernés par le ratrappage.

6. **Consultation des plannings :** Afficher le planning par date, module ou enseignant.

7. **Génération des rapports :** Imprimer ou exporter les plannings pour faciliter le travail administratif.

II.3.B.b Besoins non fonctionnels

1. **Sécurité :** Authentification sécurisée et protection des données pour éviter tout accès non autorisé.
2. **Performance :** Le système doit être rapide même avec plusieurs utilisateurs simultanés.
3. **Interopérabilité :** Capacité du système à communiquer et échanger des données avec d'autres systèmes.
4. **Fiabilité :** Les données doivent être sauvegardées sans perte pour garantir leur intégrité.
5. **Ergonomie :** Interface simple et intuitive pour tous les utilisateurs.
6. **Disponibilité :** Le système doit être accessible à tout moment.
7. **Modularité :** Le système est divisé en modules indépendants pour faciliter la maintenance.
8. **Évolutivité :** Possibilité d'ajouter de nouvelles fonctionnalités à l'avenir.
9. **Flexibilité :** Le système peut s'adapter aux changements (horaires, règles, etc.).
10. **Portabilité :** Le système fonctionne sur différentes plateformes (ordinateur, navigateur).

II.3.C Spécification des tâches

La spécification des tâches dépend de leur répartition des tâches selon les acteurs ainsi que selon les scénarios du système. C'est ce que nous voyons dans le tableau II.2, qui présente les tâches pour les acteurs.

TABLE II.2 – Identification des acteurs et description de leurs tâches

Acteurs	**Tâches**	**Sous-tâches**
Administrateur	S'identifier	
	Gérer les salles	Ajouter une salle Modifier une salle Supprimer une salle Consulter une salle

Acteurs	**Tâches**	**Sous-tâches**
	Gérer les formations	Ajouter une formation Modifier une formation Supprimer une formation Consulter une formation
	Gérer les modules	Ajouter un module Modifier un module Supprimer un module Consulter un module
	Gérer les tests	Ajouter un créneau Modifier un créneau Supprimer un créneau Consulter un créneau
	Gérer les séances de rattrapage	Ajouter une séance Modifier une séance Supprimer une séance Consulter une séance
	Gérer les demandes	Consulter les demandes Envoyer des notifications
	Gérer les profils	Ajouter un profil Modifier un profil Supprimer un profil Consulter un profil Valider un profil
	Se déconnecter	
Enseignant	S'identifier	
	S'inscrire	
	Consulter l'emploi du temps	
	Gérer les séances de rattrapage	Proposer une séance Modifier une séance Annuler une séance Consulter ses séances
	Gérer les tests	Proposer un test Modifier un test Annuler un test Consulter les tests
	Recevoir des notifications	
	Se déconnecter	
Étudiant	S'identifier	
	S'inscrire	
	Consulter les tests	

Acteurs	Tâches	Sous-tâches
	Consulter les séances de rattrapage	
	Recevoir des notifications	
	Se déconnecter	

II.3.D Diagramme de cas d'utilisation

La figure II.3 présente le diagramme de cas d'utilisation de notre système.

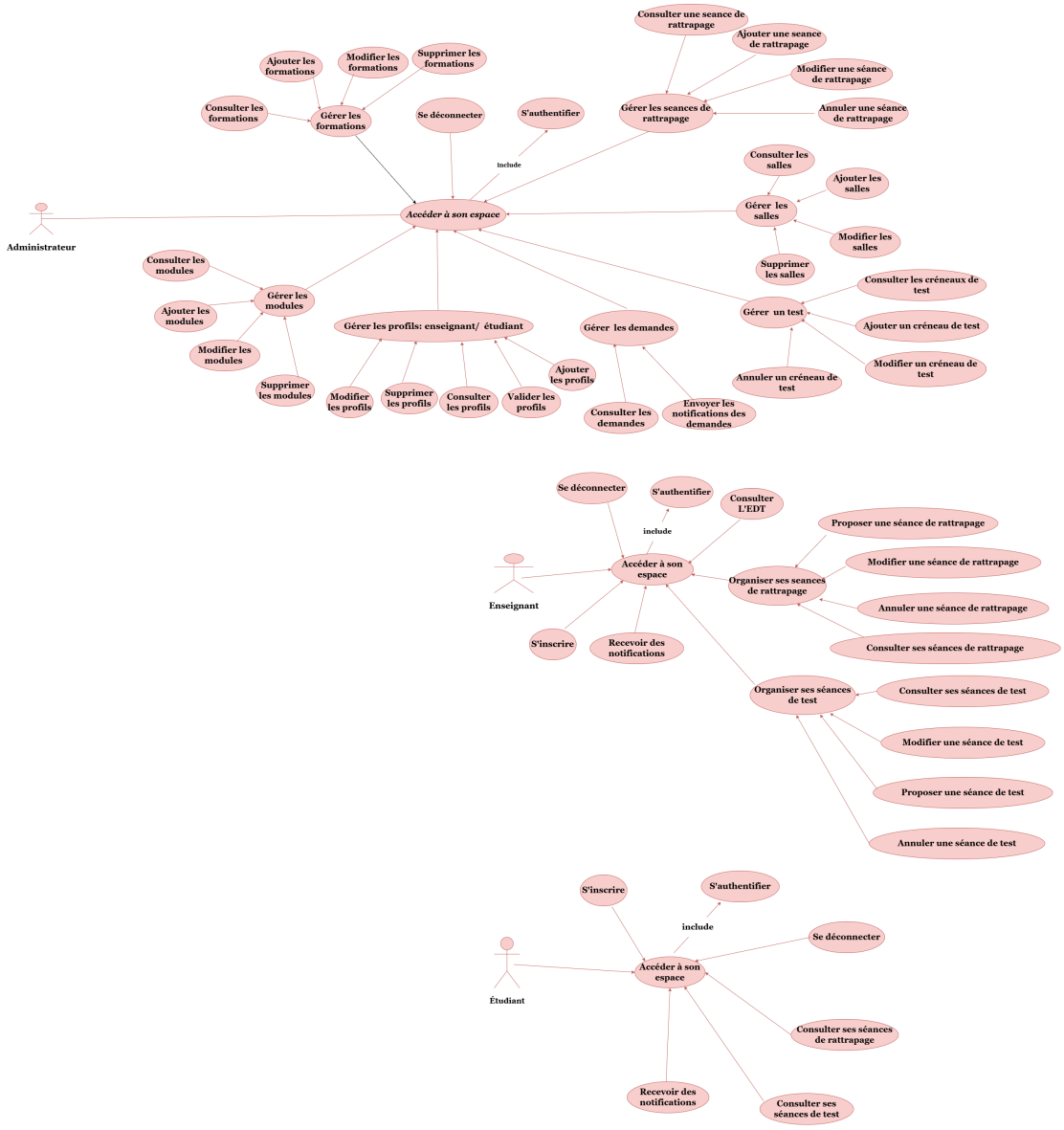


FIGURE II.3 – Diagramme de cas d'utilisation détaillé de notre système

II.4 Conception

La conception consiste à élaborer une solution concrète qui respecte les spécifications et un concept qui réponde aux besoins, tout en définissant la manière dont le programme fournit les différentes fonctionnalités requises ainsi que la description architecturale des composants (avec l'interface et les fonctionnalités), c'est ce que nous allons voir dans cette partie.

II.4.A Diagramme de séquence (DS)

Le diagramme de séquence représente la manière dont les éléments du système interagissent entre eux, au fil du temps et avec les acteurs. Dans ce qui suit, nous présentons quelques diagrammes de séquence de notre système.

II.4.A.a DS pour le cas d'utilisation "S'authentifier" en tant qu'étudiant

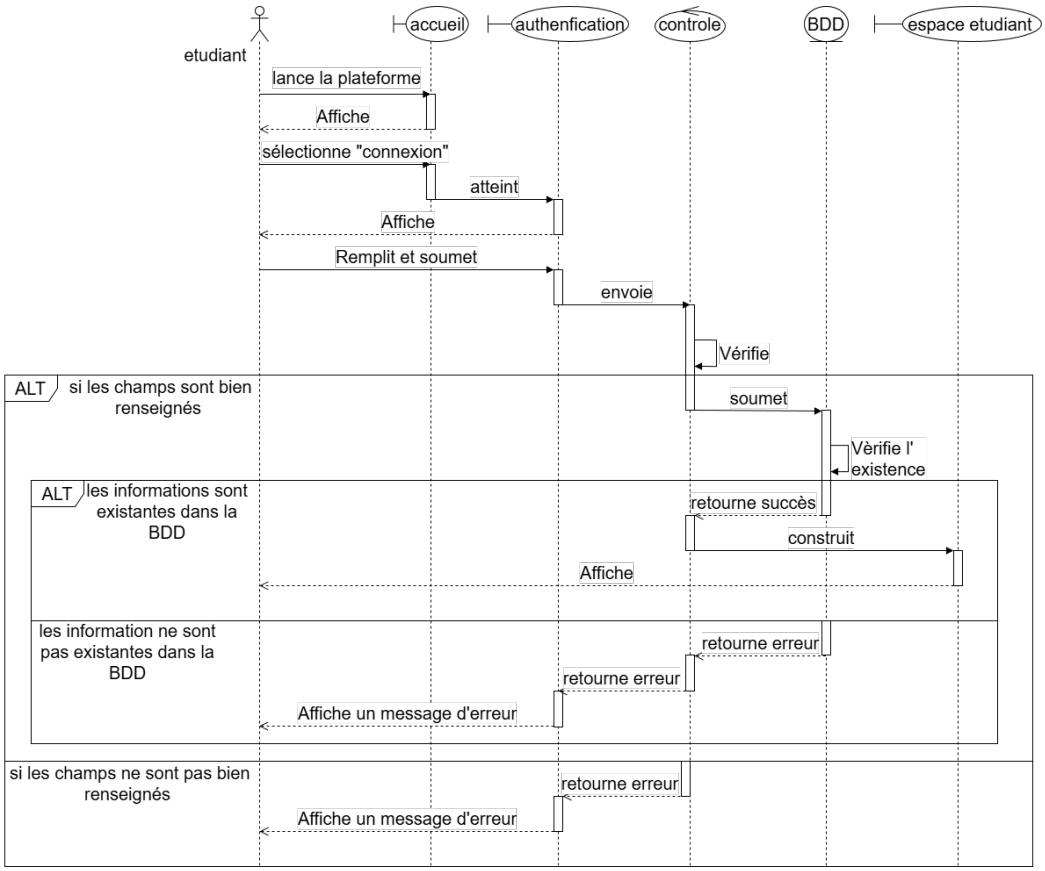


FIGURE II.4 – Diagramme de séquence pour le cas "S'authentifier" en tant qu'étudiant.

II.4.A.b DS pour le cas "Consulter les créneaux des tests" en tant qu'admin

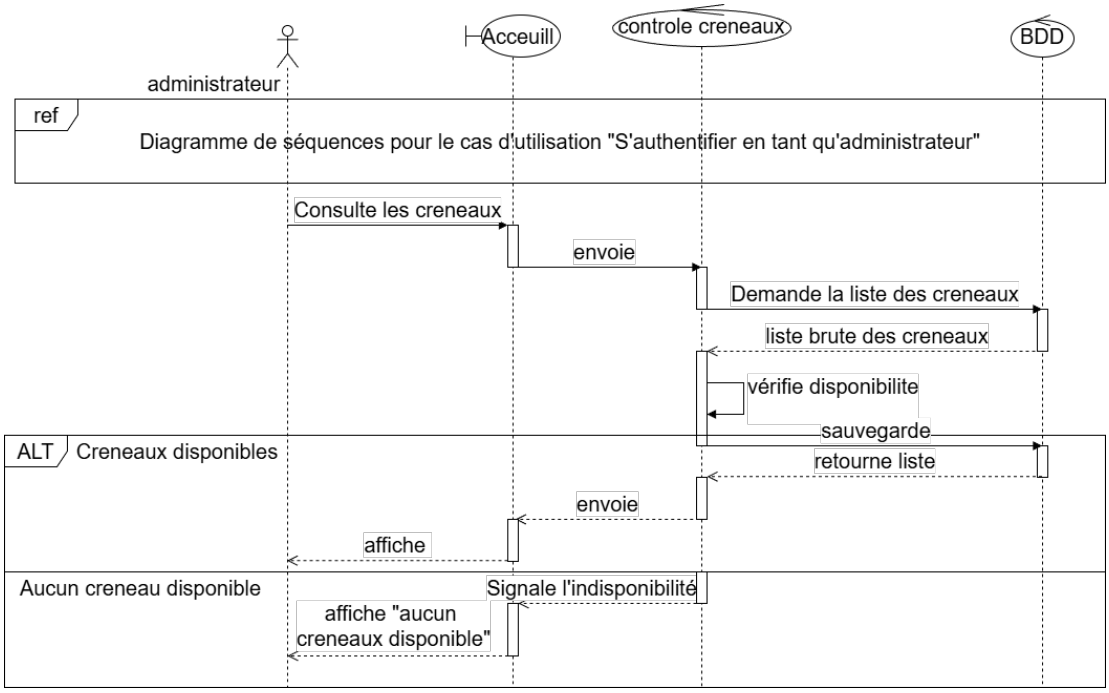


FIGURE II.5 – DS pour le cas "Consulter les créneaux des tests" en tant qu'administrateur

II.4.A.c DS pour le cas "Proposer une séance de rattrapage" en tant qu'enseignant

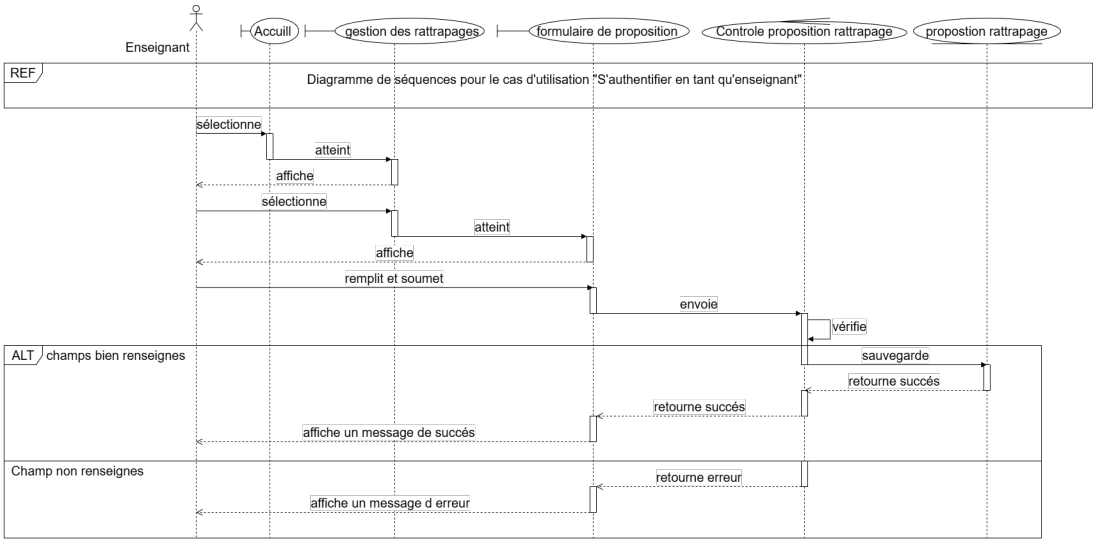


FIGURE II.6 – DS pour le cas "Proposer une séance de rattrapage" en tant qu'enseignant

II.4.B Diagramme d'activités

Le diagramme d'activités représente les règles d'enchaînement des actions et décisions au sein d'une activité. Dans ce qui suit, nous présentons quelques diagrammes d'activités de notre système.

II.4.B.a Diagramme d'activités pour le cas d'utilisation "Annuler une séance de test" en tant qu'enseignant

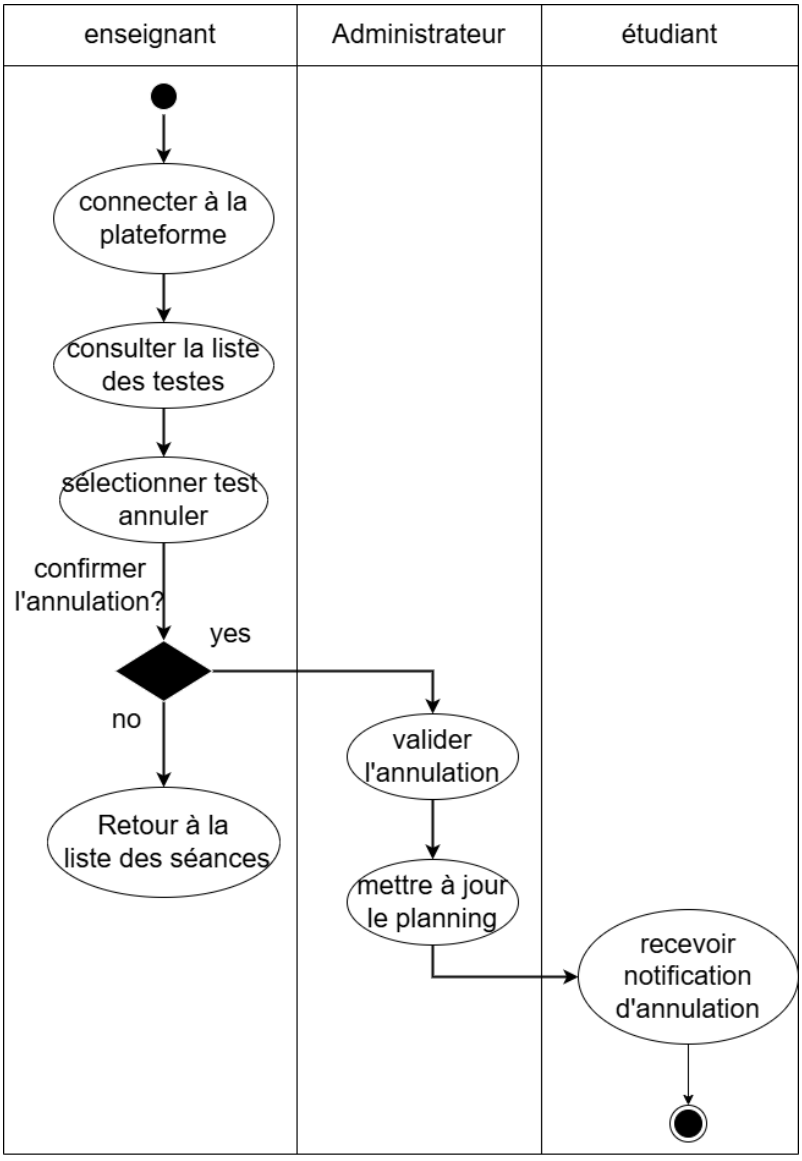


FIGURE II.7 – Diagramme d'activités pour le cas d'utilisation "Annuler une séance de test" en tant qu'enseignant

II.4.B.b Diagramme d'activités pour le cas d'utilisation "Envoyer les notifications des demandes traitées" en tant qu'enseignant

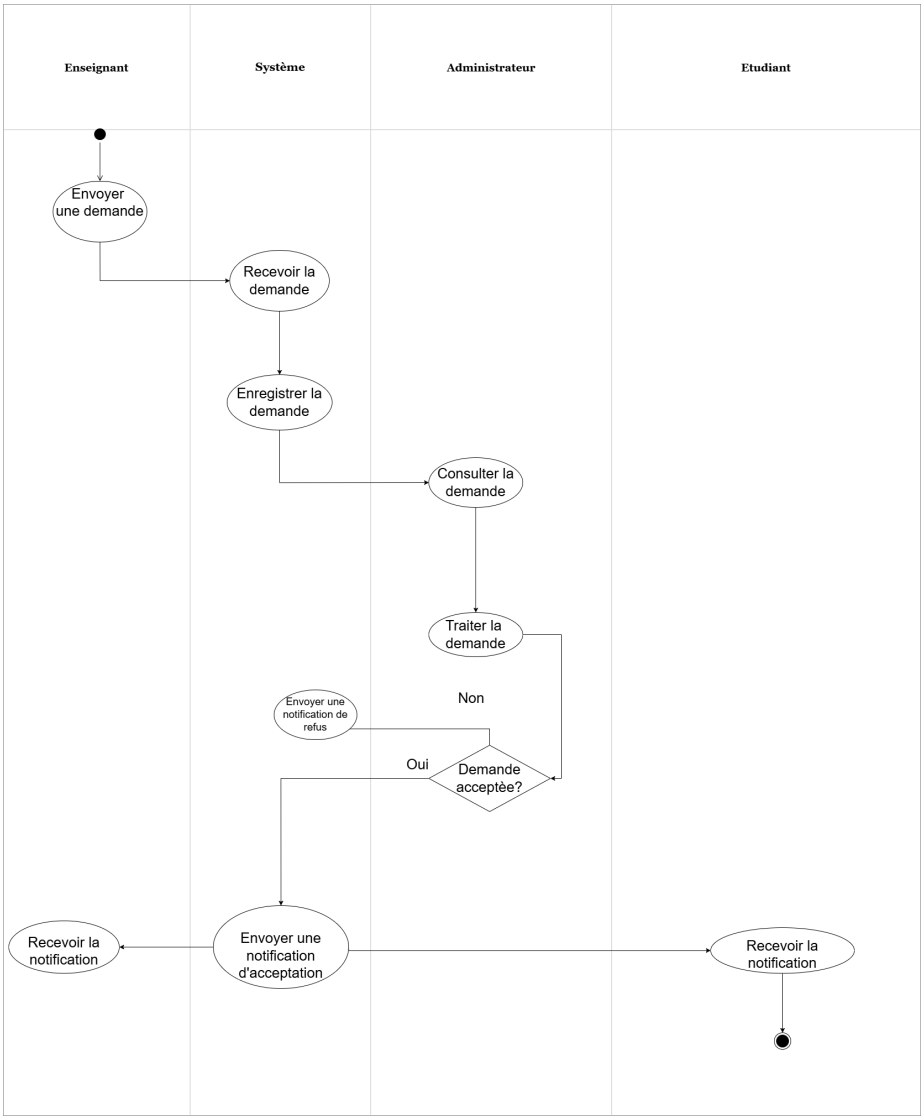


FIGURE II.8 – Diagramme d'activités pour le cas d'utilisation "Envoyer les notifications des demandes traitées" en tant qu'enseignant

II.4.C Dictionnaire des données épuré

Le tableau II.3 présente le dictionnaire des données épuré de notre système.

TABLE II.3 – Dictionnaire du données épuré de notre système

Classe	Nom d'attribut	Désignation	Type	Taille	Observation
Utilisateur	Id_Utilisateur	Identifiant utilisateur	N	10	Clé primaire
	Nom	Nom utilisateur	A	50	Obligatoire
	Email	Adresse email	AN	50	Unique
	mot_de_passe	le Mot de passe	AN	100	Obligatoire
Étudiant	Matricule	Numéro étudiant	N	10	Obligatoire
Enseignant	Spécialité	Spécialité enseignant	A	50	Obligatoire
	Grade	Grade de l'enseignant	A	50	Obligatoire
Notification	Id_Notification	Identifiant notification	N	10	Clé primaire
	Message	Contenu du message	A	255	Obligatoire
	Date	Date d'envoi	AN	10	JJ-MM-AAAA
Salle	Id_Salle	Identifiant salle	N	10	Clé primaire
	Nom_Salle	Nom salle	A	50	Obligatoire
	Bloc_Salle	Bloc salle	A	20	-
	Type_Salle	Type de salle	A	30	TD / TP
	Capacité	Capacité	N	5	-
Groupe	Id_Groupe	Identifiant groupe	N	10	Clé primaire
	Nom_Groupe	Nom groupe	A	50	Obligatoire
Séance	Id_Séance	Identifiant séance	N	10	Clé primaire
	Nom_Séance	Nom séance	A	50	-
	Heure_Debut	Heure début	AN	5	HH :MM
	Heure_Fin	Heure fin	AN	5	HH :MM
	Jour	Jour	A	20	Obligatoire
	Type	Type séance	A	30	Test / Rattrapage
	Annee_Univ	Année universitaire	AN	9	Format AAAA-AAAA
Demande	Id_Demande	Identifiant demande	N	10	Clé primaire
	Type	Type demande	A	50	Test / Rattrapage
Module	Id_Module	Identifiant module	N	10	Clé primaire
	Nom_Module	Nom module	A	50	-
	Volume_Horaire	Volume horaire	N	3	
Formation	Id_Formation	Identifiant formation	N	10	Clé primaire
	Nom_Formation	Nom formation	A	50	-
	Cycle	Cycle	A	20	Licence / Master

II.4.D Règles de gestion

1. Un utilisateur (étudiant, enseignant ou administrateur) doit s'authentifier pour accéder à son espace personnel.

2. Un étudiant peut consulter son planning et voir les séances qui le concernent.
3. Un étudiant peut recevoir des notifications envoyées par l'administrateur.
4. Un étudiant appartient obligatoirement à un seul groupe à la fois.
5. Un étudiant peut consulter les séances de rattrapage et de test qui concernent son groupe.
6. Un enseignant peut soumettre une demande de séance de rattrapage ou de test.
7. Un enseignant ne peut soumettre qu'une demande à la fois pour un module donné.
8. Un enseignant peut consulter son planning et voir les séances qui lui sont assignées.
9. Un enseignant reçoit des notifications de l'administrateur concernant ses demandes.
10. Un enseignant peut consulter l'état de ses demandes (en attente, validée, refusée).
11. L'administrateur peut créer, modifier et supprimer des salles.
12. L'administrateur peut créer, modifier et supprimer des modules.
13. L'administrateur peut créer, modifier et supprimer des formations.
14. L'administrateur peut créer, modifier et supprimer des séances (rattrapage et test).
15. L'administrateur peut valider ou rejeter les demandes soumises par les enseignants.
16. L'administrateur peut envoyer des notifications aux enseignants et aux étudiants.
17. L'administrateur peut gérer les profils des utilisateurs (enseignants et étudiants).
18. Une séance se déroule obligatoirement dans une salle et concerne un seul module.
19. Une séance de rattrapage est créée suite à la validation d'une demande d'enseignant.
20. Un test est associé à une formation et se déroule dans une séance définie.
21. Une salle ne peut accueillir qu'une seule séance à la fois pour un jour et horaire donnés.
22. Un groupe est associé à une salle et regroupe plusieurs étudiants d'une même formation.
23. L'administrateur peut consulter et modifier son propre profil.
24. L'administrateur peut accéder à un tableau de bord affichant les statistiques globales des séances, demandes et utilisateurs.

II.4.E Diagramme de classe

Le diagramme de classe est le plus important dans la modélisation orientée objet. Il représente un ensemble de classes, d'interfaces et de collaborations ainsi que leur relation. La figure II.9 présente le diagramme de classe de notre système.

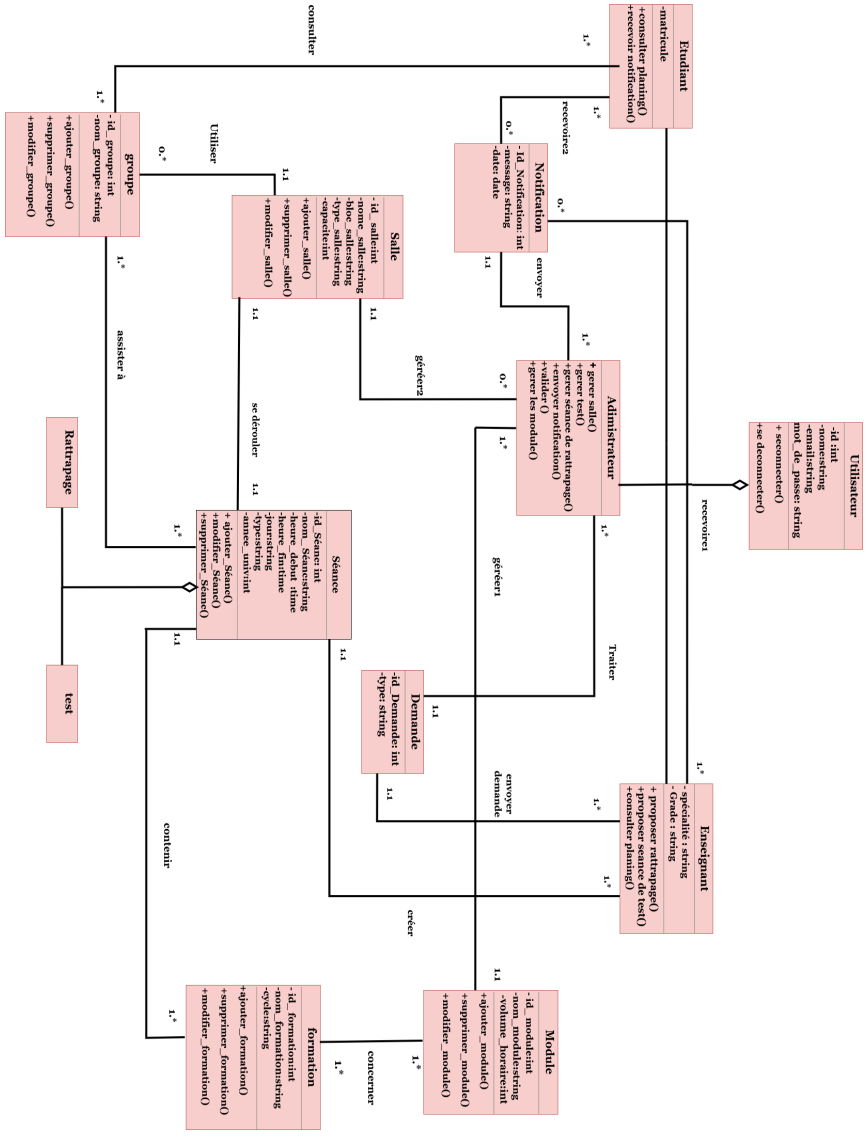


FIGURE II.9 – Diagramme de classe de notre système

II.4.F Diagramme de paquetage

Les diagrammes de paquetage sont la représentation graphique des relations existant entre les paquetages (ou espaces de noms) composant un système, dans l'UML. La figure

II.10 présente le diagramme de paquetage de notre système.

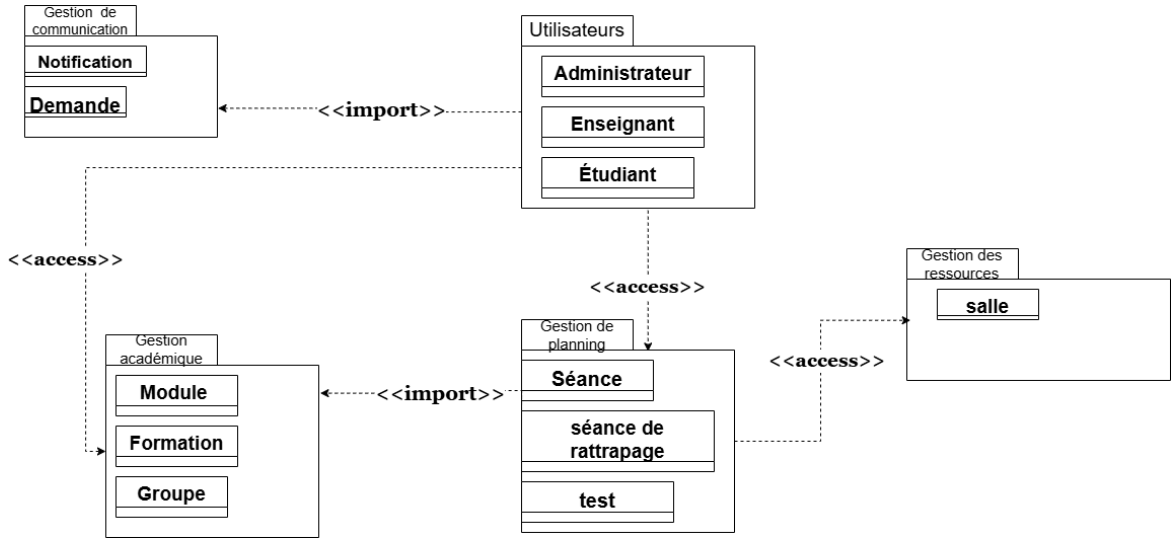


FIGURE II.10 – Diagramme de paquetage de notre système

II.4.G Diagramme de déploiement

Un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. La figure II.11 présente le diagramme de déploiement obtenu.

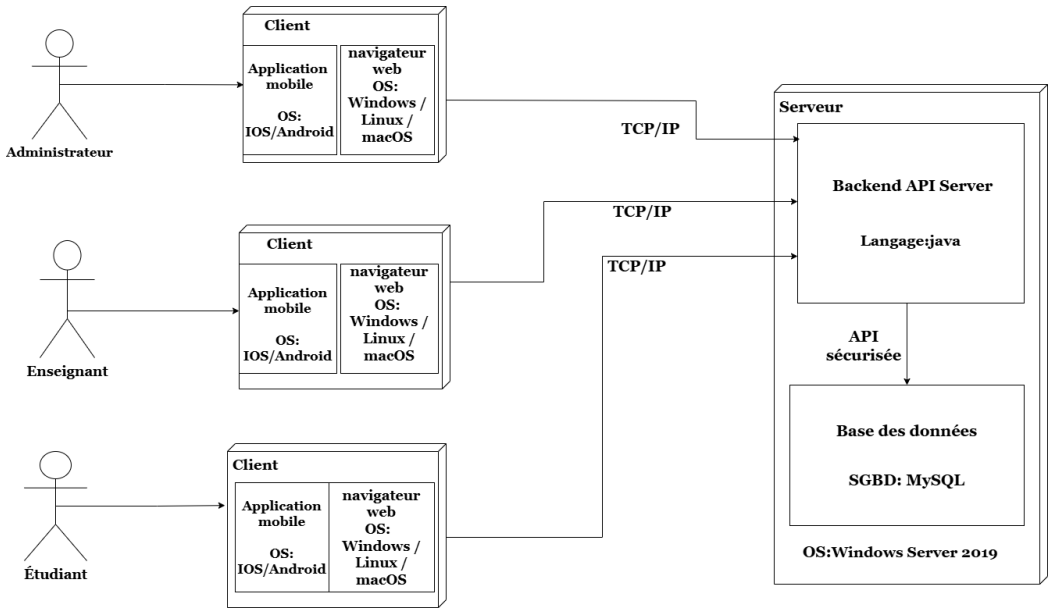


FIGURE II.11 – Diagramme de déploiement de notre système.

II.4.H Modèle Logique de Données

Le diagramme de classes a servi de base pour l'implémentation de la base de données, où chaque classe sera représentée par une table et chaque attribut par une colonne. Dans ce qui suit, nous allons présenter le MLD de notre système.
Utilisateur(id_utilisateur, Nom, Email)
Administrateur(#id_utilisateur)
Enseignant(#id_utilisateur, spécialité)
Étudiant(#id_utilisateur, matricule)
Module(id_module, nom_module, volume_horaire, #id_formation)
Formation(id_formation, nom_formation, cycle)
Salle(id_salle, nom_salle, bloc_salle, type_salle, capacité)
Groupe(id_séance, nom_séance, heure_début, heure_fin, jour, type, année_univ, #id_salle, #id_module)
Test(#id_séance)
Rattrapage(#id_séance)
Notification(id_notification, message, date, #id_utilisateur)
Demande(id_demande, type, #id_enseignant)
Séance(id_séance, nom_séance, heure_début, heure_fin, jour, type, année_univ, #id_salle, #id_module, #id_groupe, #id_enseignant)

II.4.I Modèle Physique de Données

Le Modèle Physique de Données (MPD) est l'étape où le modèle relationnel (Modèle Logique de Données) est implémenté dans un SGBD. Il consiste à transformer le modèle relationnel (Modèle Logique de Données) dans l'ordinateur en utilisant un langage de définition de données, généralement SQL dans le cas des bases de données relationnelles, tout en tenant compte des contraintes techniques. La figure II.12 illustre le modèle physique de données de notre système.

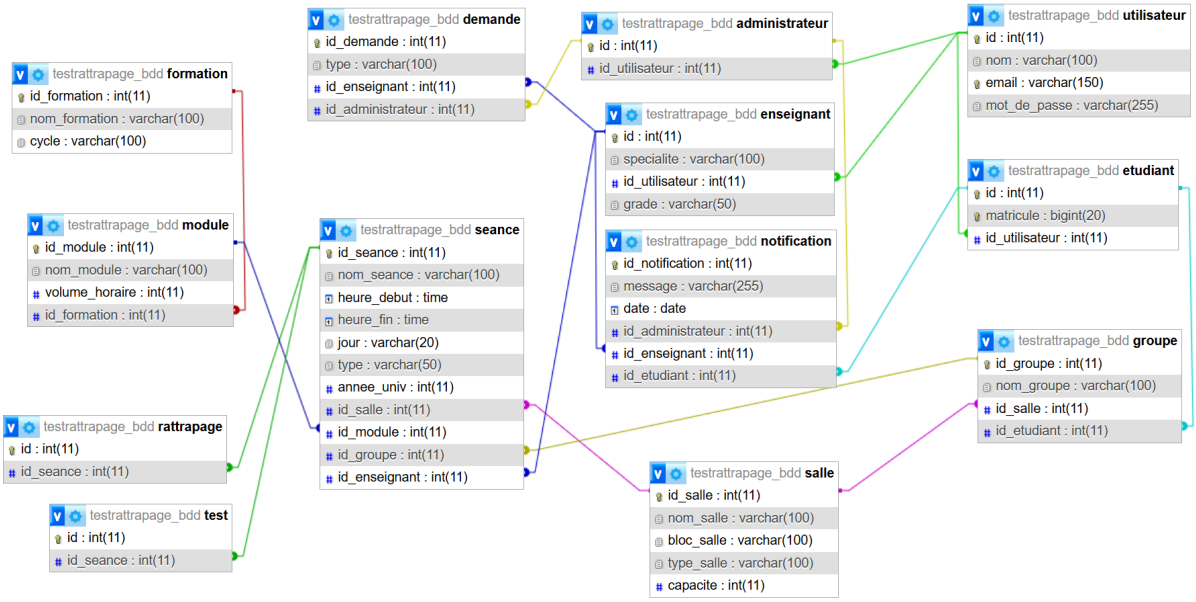


FIGURE II.12 – Modèle physique de données

II.5 Conclusion

Dans ce chapitre, les principales étapes de l'analyse et de la conception de l'application ont été présentées à travers la définition des besoins et la modélisation à l'aide des diagrammes UML. Ces éléments serviront de base pour la phase d'implémentation, qui sera détaillée dans le chapitre suivant.

Chapitre III

Réalisation

Résumé : *Dans ce chapitre, on présente la partie réalisation du système. On aborde la présentation de l'environnement logiciel utilisé pour le développement de notre plateforme, en mettant en avant les outils et logiciels de conception et de développement utilisés, ainsi que les langages de programmation et les frameworks. Enfin, on présente l'environnement matériel utilisé, qui nous a fourni une infrastructure adéquate pour la réalisation de notre projet.*

Sommaire

III.1 Introduction 27
III.2 Présentation de l'environnement logiciel utilisé 27
III.3 Présentation de l'environnement matériel utilisé 29
III.4 Conclusion 30

III.1 Introduction

Pour tout développement d'application, il est nécessaire de choisir les technologies et outils adéquats pour faciliter la réalisation. Dans ce chapitre nous allons présenter les technologies et différents outils utilisés, puis nous passerons à l'architecture du système et les outils nécessaires pour le déploiement de l'application, enfin nous allons expliquer ses fonctionnalités en présentant quelques interfaces illustratives.

III.2 Présentation de l'environnement logiciel utilisé

Pour développer notre plateforme, nous avons mis en place un ensemble d'outils couvrant le développement, la gestion des données et les tests. Cet environnement structuré nous a permis de collaborer efficacement tout en garantissant la qualité et la fiabilité du projet à chaque étape.

III.2.A Outils et logiciels de conception et de développement utilisés

Dans cette section, nous présentons les principaux outils et méthodes utilisés pour assurer le bon développement de notre plateforme, couvrant la conception, la programmation, la collaboration et la gestion du projet

III.2.A.a Draw.io



Draw.io est un outil open source lancé en 2005, permettant de créer facilement des diagrammes (UML, organigrammes, bases de données) via une interface intuitive en glisser-déposer. Sa popularité repose sur sa simplicité et son intégration avec des plateformes comme Google Drive et GitHub [9].

III.2.A.b Figma



Figma est un outil de conception et de prototypage collaboratif pour les interfaces. Lancé en 2016 par Figma, il permet de créer et partager des maquettes facilement et est largement utilisé pour la conception d'applications et de sites web [10].

III.2.A.c Visual Studio Code



Visual Studio Code est un éditeur de texte léger et performant, développé par Microsoft en 2015. Sa simplicité et sa large bibliothèque d'extensions en font un outil très apprécié pour optimiser le flux de travail et faciliter diverses tâches de développement [11].

III.2.A.d XAMPP



XAMPP est une suite de logiciels libres permettant de créer facilement un serveur local (Web, FTP et Mail),lancé en 2002 par Apache Friends. Il intègre des outils essentiels comme Apache, PHP, MySQL/MariaDB et phpMyAdmin, facilitant ainsi le développement Web et la gestion des bases de données [12].

III.2.A.e Composer



Composer est un gestionnaire de dépendances pour PHP, créé en 2012 par Nils Adermann et Jordi Boggiano. Il a été conçu pour simplifier l'installation et la gestion des bibliothèques externes, remplaçant ainsi la gestion manuelle qui rendait les projets difficiles à maintenir [13]

III.2.B Langages de programmation et frameworks utilisés

nous allons aborder les langages de programmation et les frameworks utilisés pour le développement du frontend et du backend de notre projet

III.2.B.a HTML



HTML (HyperText Markup Language) est un langage de balisage créé en 1993, conçu pour structurer et représenter les pages web. Il permet d'intégrer de l'hypertexte, des ressources multimédias (images, formulaires, etc.) grâce à un système de balises [14].

III.2.B.b CSS



Cascading Style Sheets est un langage créé en 1996 par le W3C, utilisé pour mettre en forme les pages web en définissant l'apparence des éléments HTML (couleurs, polices, mise en page) afin d'améliorer le rendu visuel des sites [15].

III.2.B.c Java Script



JavaScript est un langage de script intégré aux pages HTML, il a été lancé en 1995 . Il sert à améliorer les pages Web en exécutant des actions côté client, comme la validation des formulaires et l'affichage de boîtes de dialogue (alertes, confirmations, etc...) [16].

III.2.B.d PHP



PHP est un langage de programmation côté serveur créé en 1995, principalement utilisé pour le développement web dynamique. Il permet de traiter les données utilisateurs, interagir avec des bases de données et générer des pages web interactives. [17].

III.2.B.e SQL



SQL est un langage créé en 1974 chez IBM pour gérer les bases de données relationnelles, normalisé en 1986. Il permet de lire, ajouter, modifier et supprimer des données, et est utilisé par MySQL, PostgreSQL et Oracle [18].

III.2.B.f Bootstrap CSS



Bootstrap est un framework CSS open source créé par Twitter en 2011, qui simplifie le développement web. Il offre des composants prêts à l'emploi comme une grille responsive, des formulaires, des boutons et des extensions JavaScript optionnelles [19].

III.2.B.g Laravel



Laravel est un framework PHP open source créé en 2011, basé sur MVC, offrant une syntaxe élégante et un ORM intégré, ce qui en fait l'un des frameworks PHP les plus populaires [20].

III.3 Présentation de l'environnement matériel utilisé

Dans le cadre de la réalisation de ce projet, nous avons utilisé un ordinateur portable dont les principales caractéristiques techniques sont présentées ci-dessous.

- **Modèle :** Dell Inc Latitude 5410
- **CPU :** Intel Core i5-10310U cadencé à 1.70 GHz
- **Mémoire RAM :** 16 Go
- **Stockage :** 500 Go (163 Go utilisés)
- **GPU :** Intel UHD Graphics
- **OS :** Windows 11 64 bits

III.4 Conclusion

Dans ce chapitre, nous avons présenté les outils, technologies et interfaces graphiques utilisés pour la réalisation de notre application web. Ces éléments ont contribué à développer une application fonctionnelle avec une interface conviviale.

Conclusion générale

En conclusion, ce projet s'inscrit dans le cadre de l'amélioration de la gestion universitaire à travers le développement d'une application dédiée à la planification et à la gestion des tests ainsi que des séances de rattrapage. L'objectif principal était de proposer une solution informatique permettant d'automatiser les tâches administratives, d'optimiser l'organisation des examens et de faciliter la gestion des salles et des emplois du temps.

Pour atteindre cet objectif, nous avons mené plusieurs étapes allant de l'analyse des besoins jusqu'à la conception et au développement de l'application. Nous avons réalisé une interface permettant la gestion des examens, des séances de rattrapage, des salles ainsi que des différentes informations nécessaires au bon déroulement du processus organisationnel.

Les résultats obtenus sont satisfaisants, puisque l'application développée permet de réduire les erreurs de gestion, de gagner du temps dans l'organisation et d'améliorer l'efficacité administrative au sein des établissements universitaires. Le système mis en place offre également une meilleure fiabilité dans le traitement et la planification des données.

Ce projet nous a aussi permis de renforcer et de mettre en pratique plusieurs compétences acquises durant notre formation, notamment en analyse des besoins, conception de bases de données, programmation, gestion de projet et travail en équipe. Il représente ainsi une expérience enrichissante aussi bien sur le plan académique que professionnel.

Enfin, plusieurs perspectives d'amélioration peuvent être envisagées pour enrichir l'application, telles que l'ajout d'un système de notifications automatiques, la génération intelligente des emplois du temps, ainsi que l'intégration d'une plateforme web ou mobile afin d'améliorer davantage l'accessibilité et l'interaction des utilisateurs avec le système.

Bibliographie

[1] Infosilem. Infosilem exam – exam scheduling software. https://www.infosilem.com. Consulté en mai 2026.

[2] CELCAT. Celcat timetabler – timetabling and resource management. https://www.celcat.com. Consulté en mai 2026.

[3] UniTime. Unitime – university timetabling application. https://www.unitime.org. Consulté en mai 2026.

[4] aSc Software. asc timetables – school and university scheduling. https://www.asctimetables.com. Consulté en mai 2026.

[5] Units. Units – educational platform. https://www.units.it. Consulté en mai 2026.

[6] Pr.Gaceb. Génie logiciel (s5). Cours, miec.infodz@gmail.com. Consulté en mai 2026.

[7] T. Chaari. Cours « atelier uml ». Maître assistant, Institut Supérieur d'Électronique et de Communication. Consulté en mai 2026.

[8] Virginie Sans. Méthode merise. https://perso.univ-rennes1.fr/virginie.sans/CCI/ressources/merise.pdf. Consulté en mai 2026.

[9] JGraph Ltd. draw.io (diagrams.net). https://www.drawio.com. Consulté en mai 2026.

[10] Facem Web. Figma : définition et fonctionnement. https://facemweb.com/blog/creation-web/figma-definition/. Consulté en mai 2026.

[11] Microsoft. Visual studio code — code editing redefined. https://code.visualstudio.com. Consulté en mai 2026.

[12] Apache Friends. Xampp — apache + mariadb + php + perl. https://www.apachefriends.org/fr/index.html. Consulté en mai 2026.

[13] Nils Adermann and Jordi Boggiano. Composer — dependency manager for php, 2012. Consulté en mai 2026.

[14] W3C. Html — world wide web consortium. `https://www.w3.org/html/`. Consulté en mai 2026.

[15] W3C. Css — feuilles de style en cascade. `https://www.w3.org/Style/CSS/Overview.fr.html`. Consulté en mai 2026.

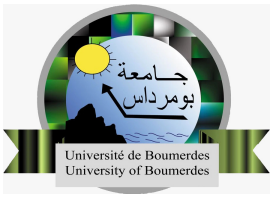
[16] W3C. Javascript — world wide web consortium. `https://www.w3.org/standards/webdesign/script`. Consulté en mai 2026.

[17] The PHP Group. Php : Hypertext preprocessor. `https://www.php.net`. Consulté en mai 2026.

[18] American National Standards Institute (ANSI). Database language sql. ANSI X3.135-1986. Consulté en mai 2026.

[19] Mark Otto and Jacob Thornton. Bootstrap — the most popular html, css, and js library. `https://getbootstrap.com`. Consulté en mai 2026.

[20] Taylor Otwell. Laravel — the php framework for web artisans. `https://laravel.com`. Consulté en mai 2026.





Département d'Informatique

Faculté des Sciences

Université M'hamed BOUGARA
BOUMERDÈS

Année académique : 2025-2026