

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université M'hamed Bougara - Boumerdès



**Mémoire de fin d'études**

Diplôme de Licence Académique

**Thème : Développement d'une application  
de gestion de l'assiduité des étudiants : cas  
du département d'informatique.**

Présenté par  
- GARITI Rania  
- LOUBAR Anes  
- SI HADJ MOHAND Nasrine

Soutenu le ...  
Devant le jury  
Encadreur : Mme Boudane  
Examineur : ...

# Dédicace

# Résumé

# Table des matières

<b>Dédicace</b>	<b>1</b>
<b>Résumé</b>	<b>2</b>
<b>1 Contexte du Travail</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 Présentation de l'organisme d'accueil . . . . .	5
1.3 Description des procédures . . . . .	6
1.4 Problématique . . . . .	6
1.5 Objectifs . . . . .	7
1.6 Conclusion . . . . .	7
<b>2 Analyse des Besoins</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Formalisation des besoins . . . . .	8
2.2.1 Les besoins fonctionnels . . . . .	8
2.2.2 Les besoins non fonctionnels . . . . .	8
2.3 Étude des acteurs . . . . .	9
2.4 Diagramme de cas d'utilisation . . . . .	9
2.5 description des cas d'utilisation . . . . .	10
2.6 Diagrammes de séquences . . . . .	14
2.6.1 Diagramme de séquence de cas d'utilisation « authentifier » . . . . .	15
2.6.2 Diagramme de séquence de cas d'utilisation « Déposer fiche de vœux » . . . . .	16
2.6.3 Diagramme de séquence de cas d'utilisation « Ajouter utilisateur » . . . . .	17
2.6.4 Diagramme de séquence de cas d'utilisation « Modifier utilisateur » . . . . .	18
2.6.5 Diagramme de séquence de cas d'utilisation « Supprimer utilisateur » . . . . .	19
2.6.6 Diagramme de séquence de cas d'utilisation « Ajouter module » . . . . .	20
2.6.7 Diagramme de séquence de cas d'utilisation « Ajouter charge » . . . . .	21
2.7 Conclusion . . . . .	21
<b>3 Conception</b>	<b>22</b>
3.1 Introduction . . . . .	22
3.2 Diagramme de classes . . . . .	22
3.2.1 Définition . . . . .	22

3.2.2	Règles de gestion . . . . .	22
3.2.3	Description des classes . . . . .	23
3.2.4	Représentation de diagramme de classe . . . . .	24
3.3	Passage au modèle relationnel . . . . .	24
3.3.1	Règles de transformation du modèle objet vers le modèle relationnel : .	24
3.3.2	Règles de transformation du modèle objet vers le modèle relationnel : .	25
3.4	Description des tables de la Base des données . . . . .	26
3.5	Conclusion . . . . .	28
<b>4</b>	<b>Implémentation</b>	<b>29</b>
4.1	Introduction . . . . .	30
4.2	Environnement de travail . . . . .	30
4.3	Interfaces graphiques . . . . .	30
4.4	Fonctionnalités principales . . . . .	30
4.5	Conclusion . . . . .	30
	<b>Bibliographie</b>	<b>31</b>


# Chapitre 1

## Contexte du Travail

### 1.1 Introduction

Dans ce chapitre, nous introduisons le contexte de notre projet. Nous allons, tout d'abord, présenter brièvement notre organisme d'accueil, à savoir le département d'informatique de la faculté des sciences de l'université M'hamed Bouguera de Boumerdes. Par la suite, nous présentons la problématique et les objectifs à atteindre par ce projet.

### 1.2 Présentation de l'organisme d'accueil

Notre étude se concentre sur le département d'informatique de la faculté des sciences de l'université M'hamed Bougara de Boumerdes  nous avons mené nos investigations et documentations afin de mettre en place une solution informatisée visant à optimiser la collecte d'informations et d'améliorer la gestion de la charge pédagogique du département. Cet établissement a été créé en 1998 pour assurer la formation de deux filières : Le DEUA et l'ingéniorat d'état dans le système classique. Depuis l'année 2004 de nouvelles formations sont apparues avec l'application du système LMD. Actuellement, le département assure la formation en informatique selon le système LMD et ingéniorat.

La figure suivante représente les différentes composantes de ce département :

- ~~— Service scolarité master~~
- ~~— Service scolarité licence~~
- ~~— Chef de département adjoint chargé de scolarité et des enseignants~~
- ~~— Responsable de Licence~~
- ~~— Responsable de Master~~
- ~~— Responsable de filière informatique~~
- ~~— Secrétariat~~
- ~~— Chef de département adjoint chargé de la poste de gradation et de la recherche~~
- ~~— Conseil scientifique du département (CSD)~~
- ~~— Chef de département~~

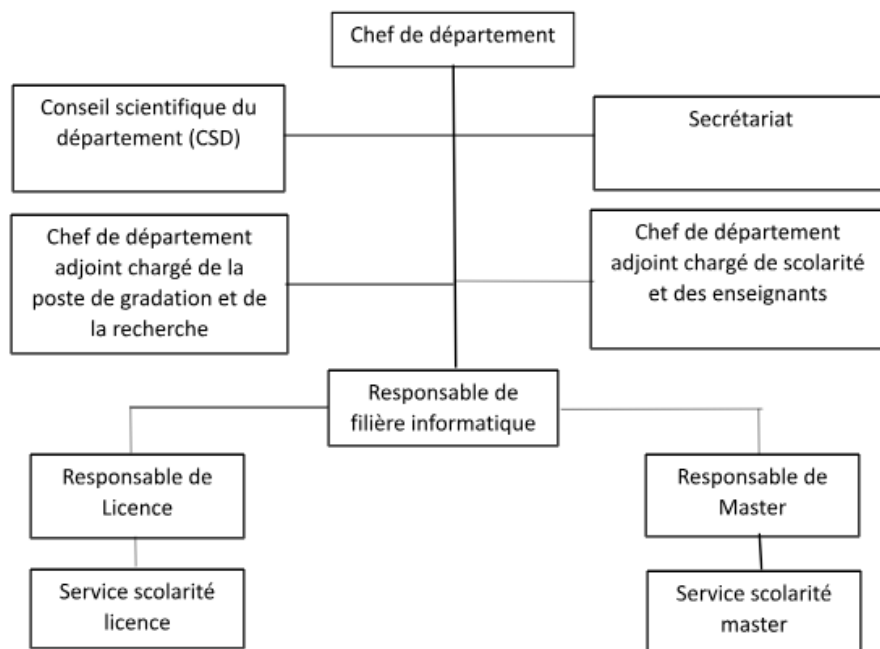


FIGURE 1.1 – Organigramme du département d'informatique

### 1.3 Description des procédures

La gestion de la charge pédagogique au sein du département d'informatique pose divers défis, notamment en ce qui concerne la planification et la distribution de la charge aux enseignants. Voici les étapes du processus :

1. L'administration propose une période pour le dépôt des fiches de vœux par les enseignants.
2. Une liste des spécialités et modules disponibles, avec les volumes horaires, est transmise.
3. Les enseignants remplissent leur fiche de vœux selon leurs priorités.
4. Collecte et étude des fiches par un responsable administratif qui propose une première affectation.
5. L'administration valide ou ajuste cette proposition selon les contraintes et statistiques.
6. Communication de la charge finale aux enseignants avec possibilité de recours.
7. Traitement des demandes de changement.
8. Ajustement final si nécessaire, puis validation définitive de la charge.

### 1.4 Problématique

Le processus actuel  présente plusieurs problèmes :

- Risque de perte de données ou de documents importants.
- Erreurs humaines dues à la saisie manuelle.
- Retards dans le traitement des fiches de vœux.

- Problèmes de communication entre enseignants et administration.
- Mauvais suivi des modifications et changements imprévus.
- Difficulté à retrouver des informations en raison du volume d’archives.

## **1.5 Objectifs**

Notre objectif est de proposer une application web capable de :

- Assurer un accès facile et continu.
- Archiver les informations pour consultation future.
- Générer des statistiques pour faciliter les décisions.
- Aider les responsables à gérer les affectations et les changements.
- Permettre aux enseignants de consulter, demander des modifications, et suivre leur charge.
- Suivre efficacement la charge pédagogique de chaque enseignant.

## **1.6 Conclusion**

Dans ce chapitre, nous avons présenté le contexte général de notre projet, les problèmes identifiés et les objectifs visés. Nous avons ainsi motivé l’intérêt d’une solution web pour gérer la charge pédagogique. Le chapitre suivant portera sur la modélisation et la conception de cette solution.



# Chapitre 2

## Analyse des Besoins

### 2.1 Introduction

Dans ce chapitre, nous aborderons la conception et la modélisation de notre solution à travers le langage de modélisation UML (Unified Modeling Language). Nous débuterons par une analyse des besoins suivie par des illustrations des diagrammes UML avec des spécifications détaillées. Cette phase permet de formaliser et clarifier les exigences fonctionnelles du système ainsi que leurs interactions avec l'utilisateur.

### 2.2 Formalisation des besoins

Cette étape a permis de saisir l'ensemble des exigences auxquelles notre application répondra.

#### 2.2.1 Les besoins fonctionnels

Les besoins fonctionnels définissent les actions que le système doit entreprendre en réponse à une requête. Notre projet vise à élaborer une application répondant aux exigences des utilisateurs. Cette application englobe les fonctionnalités essentielles pour la gestion des examens, en particulier dans le contexte du département d'informatique comme suit :

- Authentification des utilisateurs (administrateur, enseignant, responsable).
- Gestion des comptes utilisateurs (ajout, modification, suppression).
- Gestion des modules (ajout, modification, suppression).
- Gestion des groupes/sections (ajout, suppression).
- Affectation des charges pédagogiques aux enseignants.

#### 2.2.2 Les besoins non fonctionnels

Les besoins non fonctionnels définissent les objectifs relatifs aux performances du système ainsi que les contraintes de son environnement. Ces exigences techniques sont souvent formulées sous forme d'objectifs précis que le système doit atteindre. Les besoins non fonctionnels de notre application incluent :

- Sécurité :Authentification obligatoire avec matricule et mot de passe.
- Ergonomie :Interface utilisateur intuitive (formulaires clairs, navigation simplifiée).
- Performance :Temps de réponse < 2 secondes pour les opérations critiques (authentification, dépôt de vœux).
- Extensibilité :- Architecture flexible pour ajouter de nouveaux rôles (ex : chef de département)

## 2.3 Étude des acteurs

Un acteur incarne un rôle rempli par une entité externe, telle qu'une personne ou un autre système, qui interagit avec le système en question. Dans notre contexte, nous avons défini trois acteurs. Le tableau ci-dessous offre un résumé de ces acteurs et des responsabilités qu'ils assument :

Acteur	Rôle
Administrateur	<ul style="list-style-type: none"> <li>— Gérer les utilisateurs (ajout, modification, suppression).</li> <li>— Gérer les modules (ajout, suppression).</li> <li>— Superviser le système global.</li> </ul>
Enseignant	<ul style="list-style-type: none"> <li>— Déposer une fiche de vœux (choix pédagogiques).</li> <li>— Consulter sa charge pédagogique.</li> <li>— Déposer un recours si nécessaire.</li> </ul>
Responsable	<ul style="list-style-type: none"> <li>— Affecter les charges pédagogiques aux enseignants.</li> <li>— Traiter les fiches de vœux et les recours.</li> <li>— <del>Gérer les groupes/sections.</del></li> </ul>

TABLE 2.1 – Rôles des acteurs dans le système

## 2.4 Diagramme de cas d'utilisation

Le but de ce diagramme est de fournir une vue d'ensemble des interfaces du logiciel à venir. Il permet d'illustrer les fonctionnalités du système ainsi que les attentes des utilisateurs à son égard. Il inclut un ensemble d'acteurs qui interagissent avec les cas d'utilisation de l'application, et ainsi, présenter les objectifs que le système ou l'application doit atteindre.

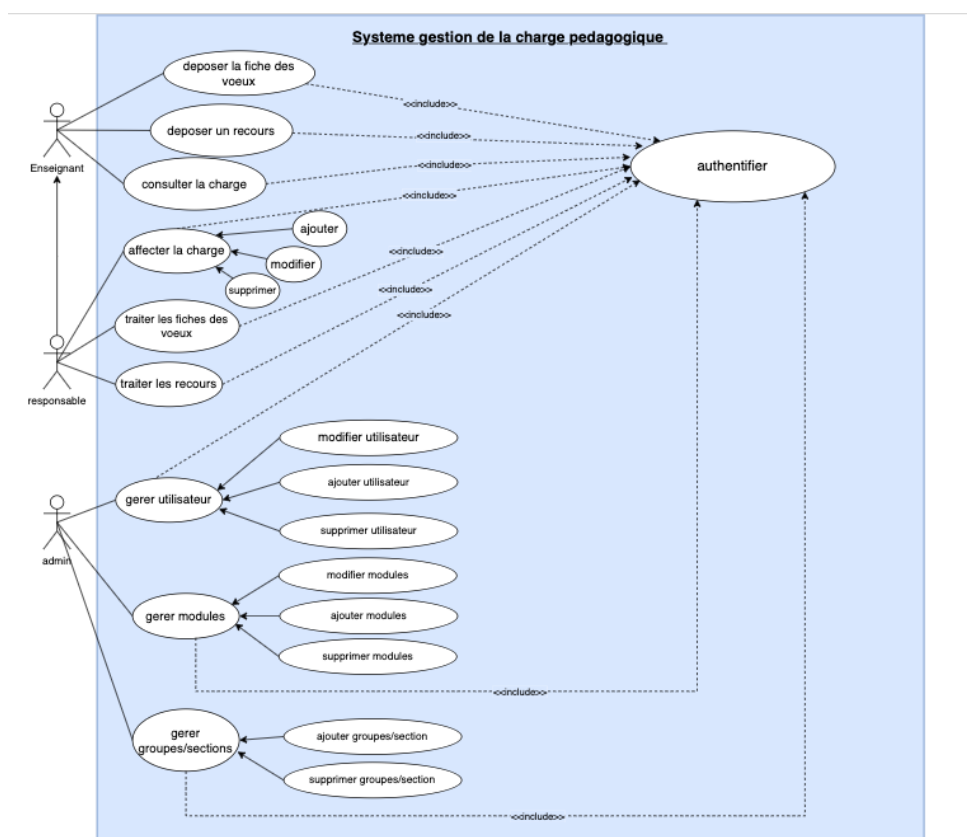


FIGURE 2.1 – Diagramme de cas d'utilisation

## 2.5 description des cas d'utilisation

Nous décrivons, dans cette section, d'une façon détaillée quelques cas d'utilisation. Il s'agit de donner tous les scénarios possibles.

### — Cas d'utilisation :Authentifier

<b>Titre</b>	Authentifier
<b>Acteurs</b>	Administrateur, Enseignant, ...
<b>Objectif</b>	Permettre à un utilisateur de s'identifier de manière sécurisée pour accéder à son espace personnel.
<b>Préconditions</b>	Disposer d'un matricule et d'un mot de passe valides.
<b>Postconditions</b>	Accès à l'espace utilisateur selon le profil.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page d'authentification.</li> <li>2. Le système affiche un formulaire de connexion.</li> <li>3. L'utilisateur saisit ses identifiants puis clique sur « Valider ».</li> <li>4. Le système vérifie les informations.</li> <li>5. L'utilisateur est redirigé vers son espace personnel.</li> </ol>
<b>Scénarios alternatifs</b>	<p>A1) En 4 : si l'un des identifiants est incorrect :</p> <p>A1.1) Le système affiche « Mot de passe ou matricule incorrect » et retourne à l'étape 2.</p>

<b>Titre</b>	<del>Authentifier</del>
<b>Acteurs</b>	<del>Administrateur, Enseignant, ...</del>
<b>Objectif</b>	<del>Permettre à un utilisateur de s'identifier de manière sécurisée pour accéder à son espace personnel.</del>
<b>Préconditions</b>	<del>Disposer d'un matricule et d'un mot de passe valides.</del>
<b>Postconditions</b>	<del>Accès à l'espace utilisateur selon le profil.</del>

TABLE 2.2: Description textuelle du cas "Authentifier".

— Cas d'utilisation : Déposer fiche de vœux

<b>Titre</b>	Déposer fiche de vœux
<b>Acteurs</b>	Enseignant
<b>Objectif</b>	Permettre à l'enseignant de remplir puis déposer sa fiche de vœux.
<b>Préconditions</b>	Être authentifié en tant qu'enseignant.
<b>Postconditions</b>	La fiche de vœux est déposée.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. L'enseignant clique sur "Fiche de vœux".</li> <li>2. Le système affiche la page "Fiche de vœux".</li> <li>3. L'enseignant saisit trois choix obligatoires et deux choix optionnels, puis valide.</li> <li>4. Le système vérifie les informations saisies.</li> <li>5. Un message de confirmation est affiché.</li> <li>6. L'enseignant confirme l'envoi.</li> <li>7. Le système affiche un message de succès.</li> </ol>
<b>Scénarios alternatifs</b>	<p>B1) En 4 : si la fiche est mal remplie :</p> <p>B1.1) Le système affiche un message d'erreur et retourne à l'étape 2.</p> <p>B2) Si la fiche a déjà été déposée :</p> <p>B2.1) Le système affiche « Fiche déjà envoyée. Voulez-vous la modifier ? » et retourne à l'étape 3.</p>
<b>Extensions</b>	Aux étapes 3 et 5, l'enseignant peut annuler le processus de dépôt.

TABLE 2.3: Description textuelle du cas "déposer fiche de vœux".

— Cas d'utilisation : Ajouter utilisateur

<b>Titre</b>	Ajouter utilisateur
<b>Acteurs</b>	Admin
<b>Objectif</b>	Permettre à l'administration d'ajouter des utilisateurs.
<b>Préconditions</b>	Être authentifié comme admin.
<b>Postconditions</b>	Les données de l'utilisateur sont bien ajoutées.

<b>Scénario principal</b>	1. L'admin accède à la page de gestion des utilisateurs. 2. Le système affiche la page correspondante. 3. L'admin remplit le formulaire d'ajout d'utilisateur. 4. Il clique sur le bouton « Ajouter ». 5. Le système vérifie les informations et demande confirmation. 6. L'admin confirme. 7. Les données sont enregistrées et un message de succès est affiché.
<b>Scénarios alternatifs</b>	E1) En 5 : formulaire incomplet. 7') Message d'erreur et retour à l'étape 2. E2) En 5 : matricule déjà existant. 7'') Message « Ce matricule existe », retour à l'étape 2.
<b>Extensions</b>	En 3 ou 5) l'admin peut annuler l'opération.

TABLE 2.4: Description textuelle du cas “ajouter utilisateur”.

## — Cas d'utilisation : Modifier utilisateur

<b>Titre</b>	Modifier utilisateur
<b>Acteurs</b>	Admin
<b>Objectif</b>	Permettre à l'administration de modifier les données d'un utilisateur.
<b>Préconditions</b>	Être authentifié comme admin.
<b>Postconditions</b>	Les données sont modifiées.
<b>Scénario principal</b>	1. L'admin accède à la page de gestion des utilisateurs. 2. Le système affiche la page. 3. L'admin recherche l'utilisateur. 4. Il sélectionne l'utilisateur et clique sur « Modifier ». 5. Le système affiche un formulaire prérempli. 6. L'admin modifie les champs nécessaires et clique sur « Modifier ». 7. Le système vérifie les données et demande confirmation. 8. L'admin confirme. 9. Les données sont enregistrées et un message de succès est affiché.
<b>Scénarios alternatifs</b>	E1) En 6 : formulaire mal rempli. 7') Message d'erreur, retour à l'étape 2. E2) En 7 : matricule déjà existant. 9') Message « Ce matricule existe », retour à l'étape 2.
<b>Extensions</b>	En 5 et 6, possibilité d'annuler l'opération.

TABLE 2.5: Description textuelle du cas “modifier utilisateur”.

## — Cas d'utilisation : Supprimer utilisateur

<b>Titre</b>	Supprimer utilisateur
<b>Acteurs</b>	Admin
<b>Objectif</b>	Permettre à l'administration de supprimer un utilisateur.
<b>Préconditions</b>	Être authentifié comme admin.
<b>Postconditions</b>	L'utilisateur est supprimé du système.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. L'admin accède à la page de gestion des utilisateurs.</li> <li>2. Le système affiche la page.</li> <li>3. L'admin recherche l'utilisateur.</li> <li>4. Le système affiche les résultats.</li> <li>5. L'admin sélectionne et clique sur « Supprimer ».</li> <li>6. Le système demande confirmation.</li> <li>7. Les données sont supprimées et un message de succès est affiché.</li> </ol>
<b>Scénarios alternatifs</b>	E1) En 3 : critère inexistant. 4') Message d'erreur, retour à l'étape 3.
<b>Extensions</b>	En 6) l'admin peut annuler l'opération.

TABLE 2.6: Description textuelle du cas "supprimer utilisateur"

## — Cas d'utilisation : Ajouter module

<b>Titre</b>	Ajouter module
<b>Acteurs</b>	Admin
<b>Objectif</b>	Permettre à l'admin d'ajouter un module.
<b>Préconditions</b>	Être authentifié comme admin.
<b>Postconditions</b>	Le module est ajouté au système.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. L'admin accède à la page de gestion des modules.</li> <li>2. Le système affiche la page.</li> <li>3. L'admin remplit le formulaire du module.</li> <li>4. Il clique sur « Ajouter ».</li> <li>5. Le système vérifie et demande confirmation.</li> <li>6. L'admin confirme.</li> <li>7. Le module est enregistré et un message de succès est affiché.</li> </ol>
<b>Scénarios alternatifs</b>	E1) En 5 : champ obligatoire vide. 7') Message d'erreur, retour à l'étape 2. E2) En 5 : matricule de module déjà existant. 7'') Message « Ce matricule existe », retour à l'étape 2.
<b>Extensions</b>	En 3 ou 5) l'admin peut annuler l'opération.


TABLE 2.7: Description textuelle du cas "ajouter module"

## — Cas d'utilisation : Ajouter charge

<b>Titre</b>	Ajouter charge
<b>Acteurs</b>	Responsable
<b>Objectif</b>	Permettre au responsable d'ajouter une charge pour un enseignant.
<b>Préconditions</b>	Être authentifié comme responsable.
<b>Postconditions</b>	La charge est bien affectée.
<b>Scénario principal</b>	<ol style="list-style-type: none"> <li>1. Le responsable accède à la page de gestion de la charge.</li> <li>2. Le système affiche la page.</li> <li>3. Le responsable choisit « Ajouter charge ».</li> <li>4. Il sélectionne un enseignant (liste ou critères).</li> <li>5. Le système affiche un formulaire.</li> <li>6. Le responsable remplit les champs.</li> <li>7. Il clique sur « Affecter ».</li> <li>8. Le système vérifie et demande confirmation.</li> <li>9. Le responsable confirme.</li> <li>10. La charge est enregistrée et un message de succès est affiché.</li> </ol>
<b>Scénarios alternatifs</b>	<p>E1) En 4 : champs obligatoires vides. 5') Message d'erreur, retour à l'étape 4.</p> <p>E2) En 8 : dépassement de charge (&gt;18h). 8') Message d'erreur.</p>
<b>Extensions</b>	En 5 : si la charge de l'enseignant est complète, possibilité d'annuler.

TABLE 2.8: Description textuelle du cas "ajouter charge"

## 2.6 Diagrammes de séquences

Un diagramme de séquence est une représentation visuelle de l'interaction entre différents objets ou composants dans un système logiciel. Il met en évidence la séquence chronologique des messages échangés entre ces objets pour accomplir un certain comportement ou une fonctionnalité du système .

### 2.6.1 Diagramme de séquence de cas d'utilisation « authentifier »

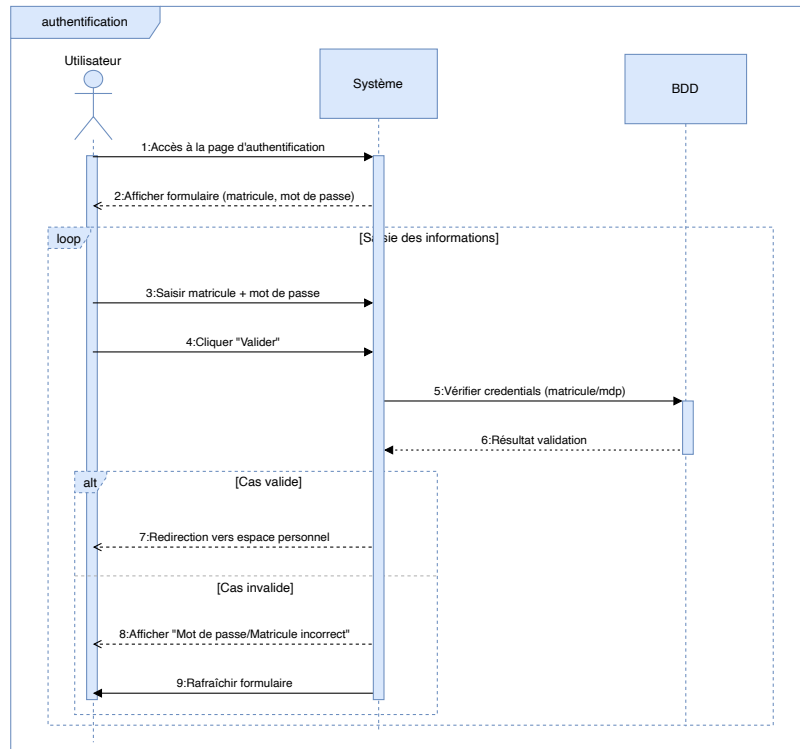


FIGURE 2.2 – Diagramme de séquence de l'authentification.



## 2.6.2 Diagramme de séquence de cas d'utilisation « Déposer fiche de vœux »

»

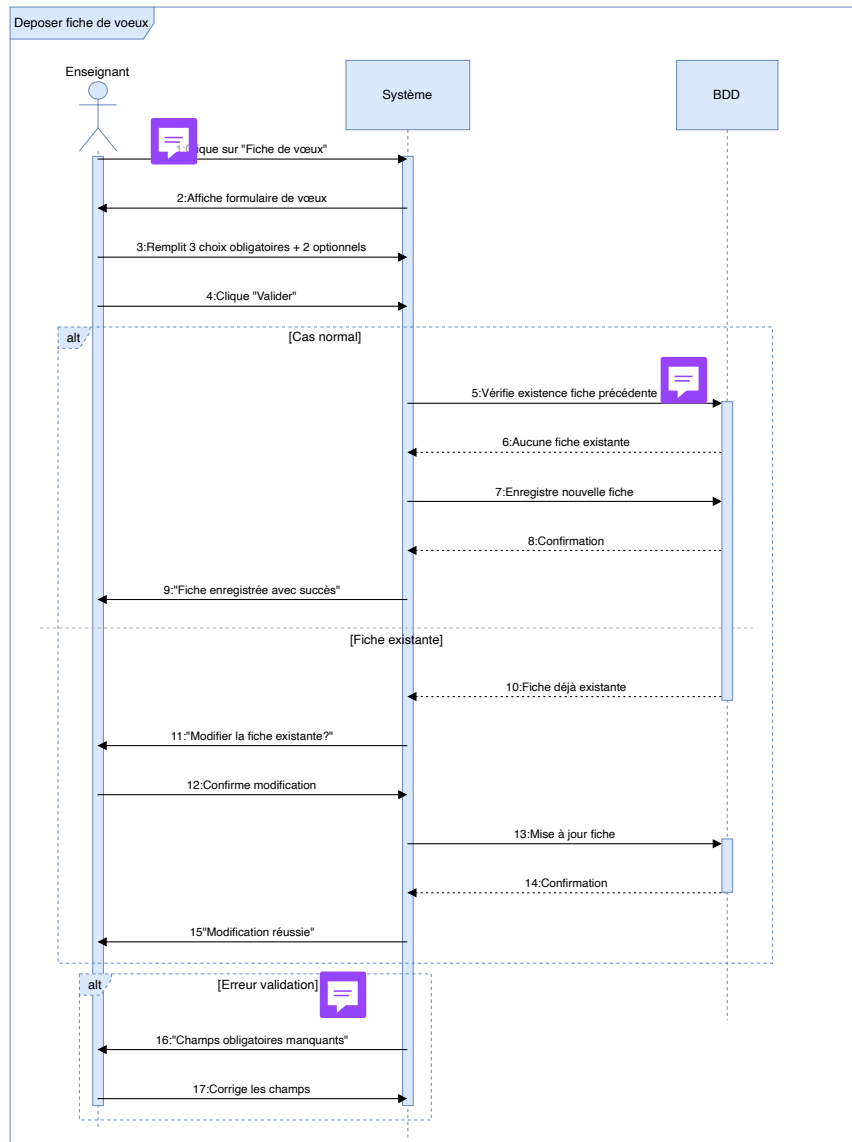


FIGURE 2.3 – Diagramme de séquence de Déposer fiche de vœux.

### 2.6.3 Diagramme de séquence de cas d'utilisation « Ajouter utilisateur »

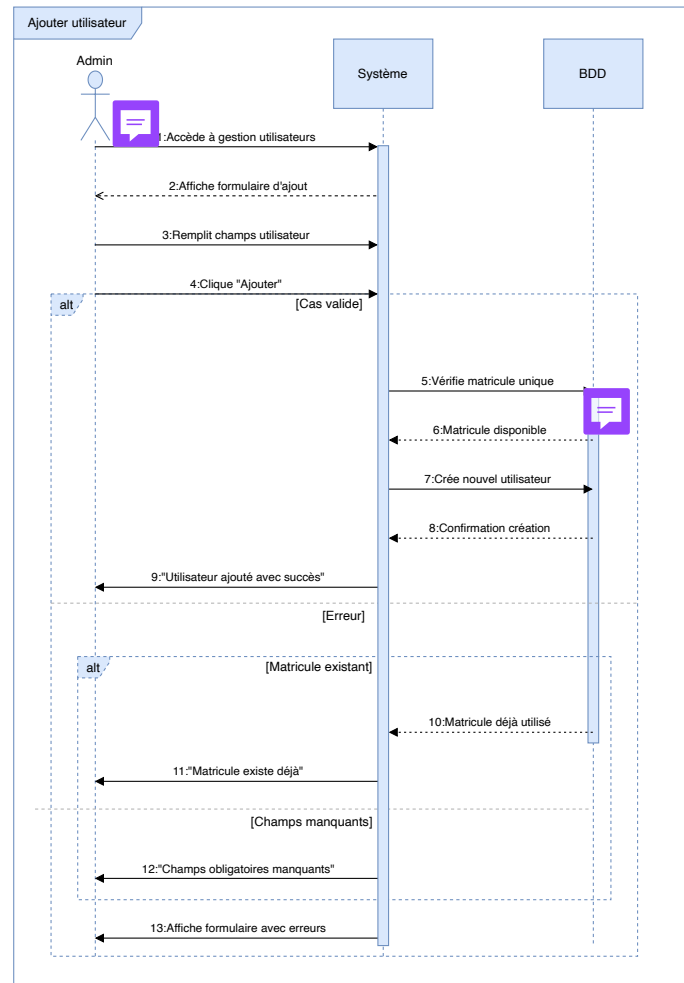


FIGURE 2.4 – Diagramme de séquence d’Ajouter utilisateur.

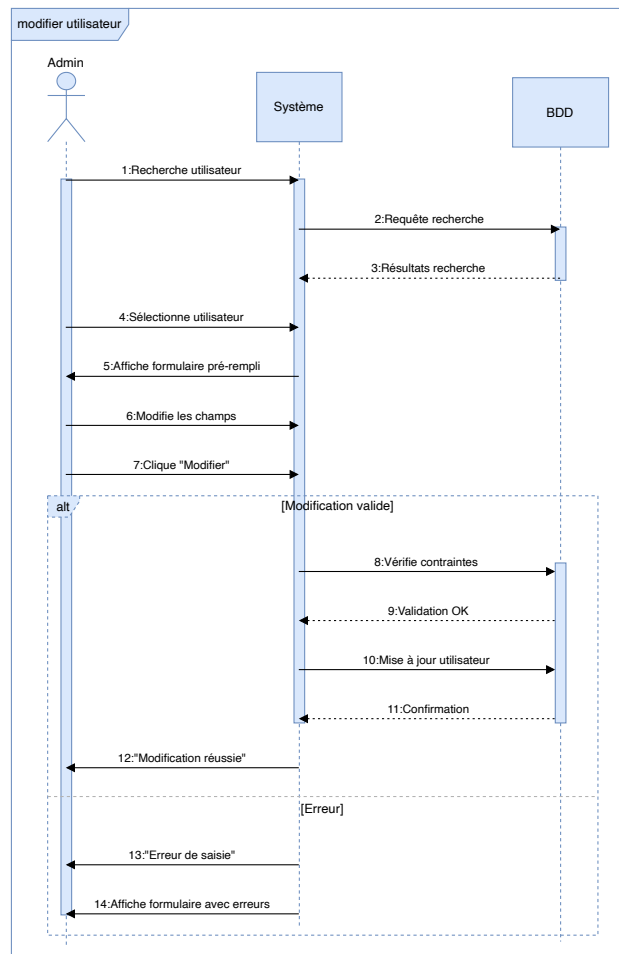
**2.6.4 Diagramme de séquence de cas d'utilisation « Modifier utilisateur »**

FIGURE 2.5 – Diagramme de séquence de Modifier utilisateur.

**2.6.5 Diagramme de séquence de cas d'utilisation « Supprimer utilisateur »**

»

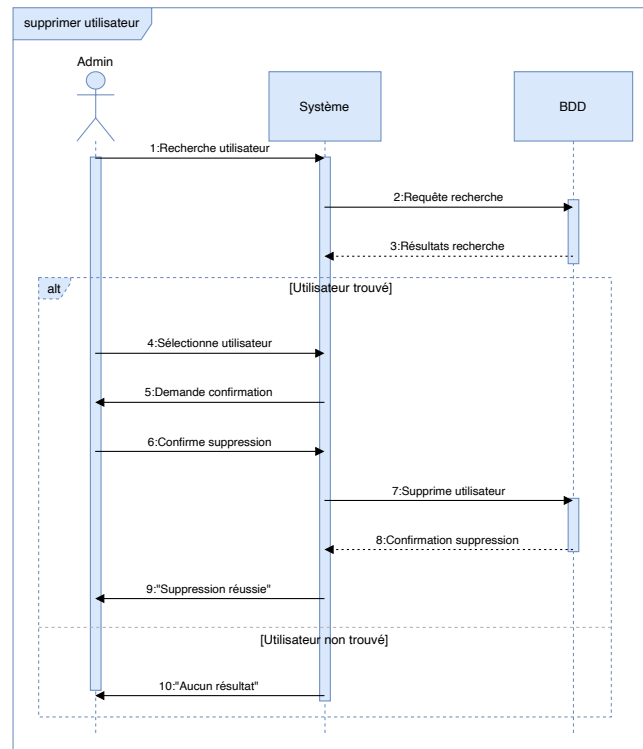


FIGURE 2.6 – Diagramme de séquence de Supprimer utilisateur

## 2.6.6 Diagramme de séquence de cas d'utilisation « Ajouter module »

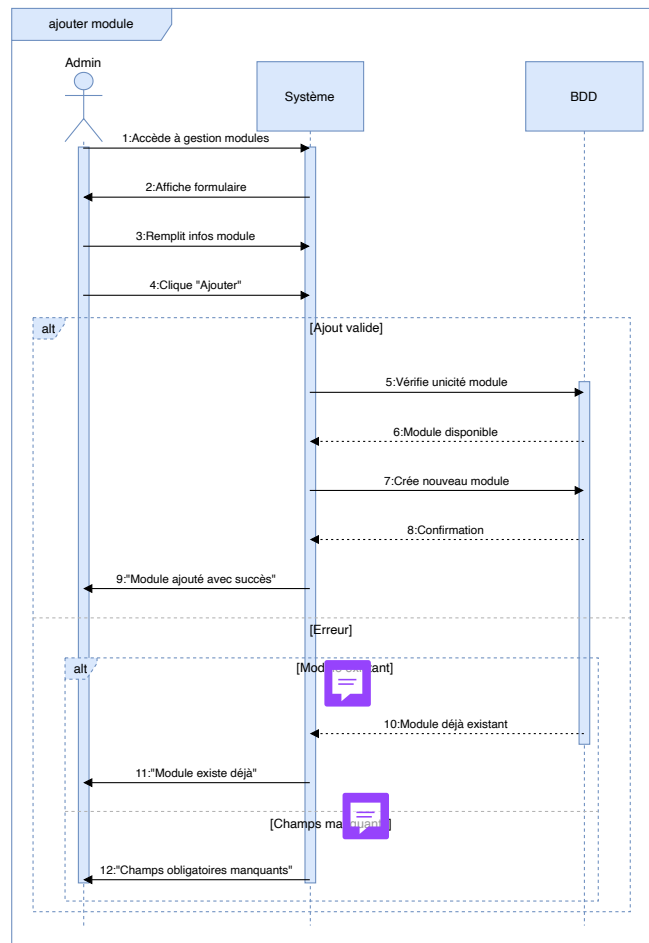


FIGURE 2.7 – Diagramme de séquence de Ajouter module.

### 2.6.7 Diagramme de séquence de cas d'utilisation « Ajouter charge »

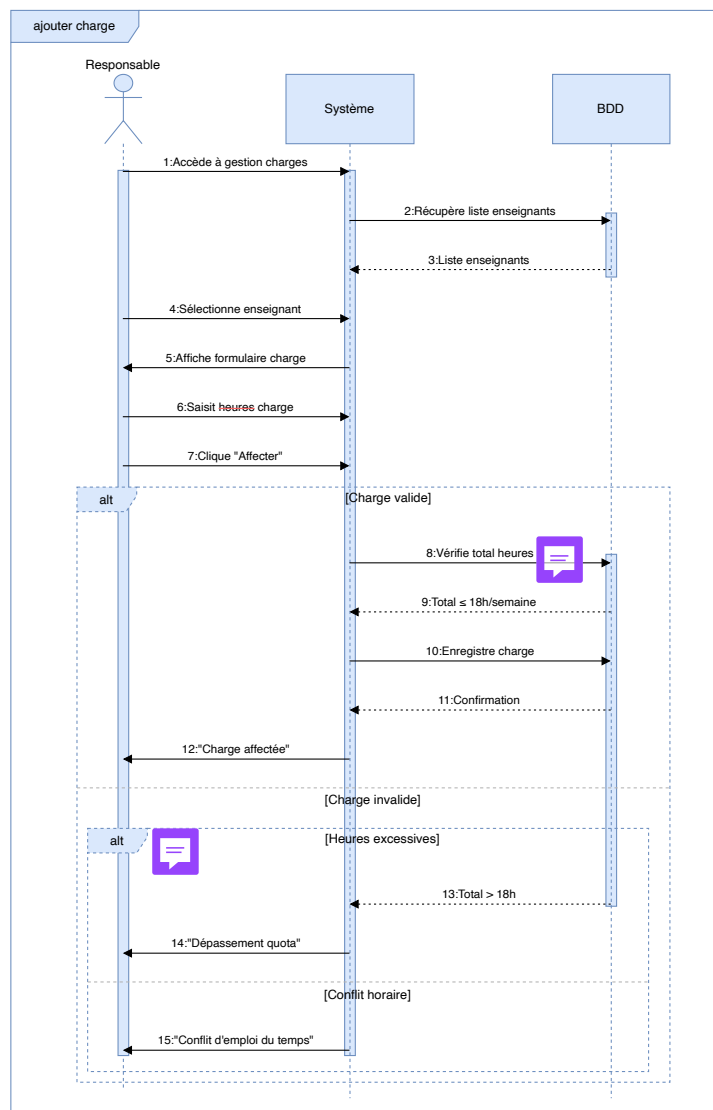


FIGURE 2.8 – Page 2 of the PDF as a figure.

## 2.7 Conclusion

Ce chapitre se concentre sur la modélisation de notre application ~~en abordant~~ en se fondant sur l'analyse des données. Nous avons utilisé un ensemble spécifique de diagrammes UML, y compris le diagramme de cas d'utilisation et le diagramme de séquence. Le chapitre suivant traitera de la représentation de la structure statique de notre application à travers le diagramme de classes et le modèle relationnel.

# Chapitre 3

## Conception

### 3.1 Introduction

Dans ce chapitre, nous structurons la conception de notre système, l'interaction, l'architecture des composants et leurs caractéristiques, ~~qui servira de référence pour le développement de l'application afin de garantir la cohérence de l'architecture du système avant de passer à l'implémentation.~~

### 3.2 Diagramme de classes

#### 3.2.1 Définition

Un diagramme de classe est une représentation visuelle statique qui montre la structure et les relations entre les différentes parties d'un système logiciel. Il illustre les différentes classes, leurs caractéristiques, leurs fonctions et leurs liens, ainsi que les interfaces et les interdépendances entre elles.

#### 3.2.2 Règles de gestion

Pour réaliser le diagramme de classe de notre système, nous avons d'abord identifié les règles de gestion du système. À partir de ces règles, nous avons identifié les différentes classes, attributs et méthodes ainsi que les relations entre ces classes. Dans ce qui suit, nous citons les règles de gestion que nous avons utilisées :

- R1 : Chaque enseignant peut déposer une fiche de vœux pour exprimer ses préférences de charge pédagogique (cours, TD, TP).
- R2 : Chaque enseignant peut déposer un recours pour ~~contester une note de contrôle continu ou d'examen~~ durant la période définie.
- R3 : Seul le responsable ~~(chef de département)~~ peut traiter les fiches de vœux et effectuer l'affectation de la charge pédagogique.
- R4 : Seul le responsable peut consulter la charge pédagogique agrégée par module, section, groupe et enseignant.

- R5 : L'administrateur peut gérer les comptes utilisateurs, les modules, les groupes et les sections, avec des opérations d'ajout, de modification et de suppression.
- R6 : Toute action de gestion (dépôt de vœu, dépôt de recours, traitement, modification, suppression) nécessite une authentification préalable.
- U7 : U7 : Chaque module possède un tableau de priorité d'enseignants, ordonné par grade et années d'expérience sur ce module.
- R8 : Si un groupe possède déjà un chargé pour un type d'enseignement (cours, TD ou TP) sur un module donné, la case à cocher correspondante n'est pas affichée dans l'interface de dépôt de vœux.
- R9 : Si tous les groupes d'une section ont déjà un chargé pour un type d'enseignement donné sur un module, la case à cocher de ce type n'apparaît plus pour tous les enseignants.
- R10 : Les interfaces de gestion (Fiche de vœux, Charge, Recours) affichent dynamiquement les modules, niveaux, spécialités, sections et groupes selon la configuration actuelle du système.
- R11 : Les périodes d'ouverture et de fermeture pour le dépôt des fiches de vœux et des recours sont configurables et gérées par l'administrateur via des dates de début et de fin.
- R12 : Les statistiques (nombres d'étudiants, d'enseignants par grade, disponibilités) sont calculées en temps réel et accessibles depuis l'interface de l'administrateur et du responsable.

### 3.2.3 Description des classes

- **Utilisateur** : Contient les informations communes à tous les utilisateurs du système. (ses attributs et ses méthodes). C'est la classe mère des classes Enseignant et Responsable.
- **Enseignant** : Hérite de Utilisateur. Contient des informations propres aux enseignants.
- **Responsable** : Hérite de Utilisateur. Contient l'année de début du poste de responsable.
- **Spécialité** : Représente une filière d'études.
- **Module** : Représente une unité d'enseignement (ses attributs et ses méthodes).
- **Charge** : Représente une période de charge d'enseignement. (ses attributs et ses méthodes).
- **FicheDeVoeux** : Permet aux enseignants de déposer leurs souhaits de charge. (ses attributs et ses méthodes)
- **Recours** : Gère les demandes de révision. (ses attributs et ses méthodes).



### 3.2.4 Représentation de diagramme de classe

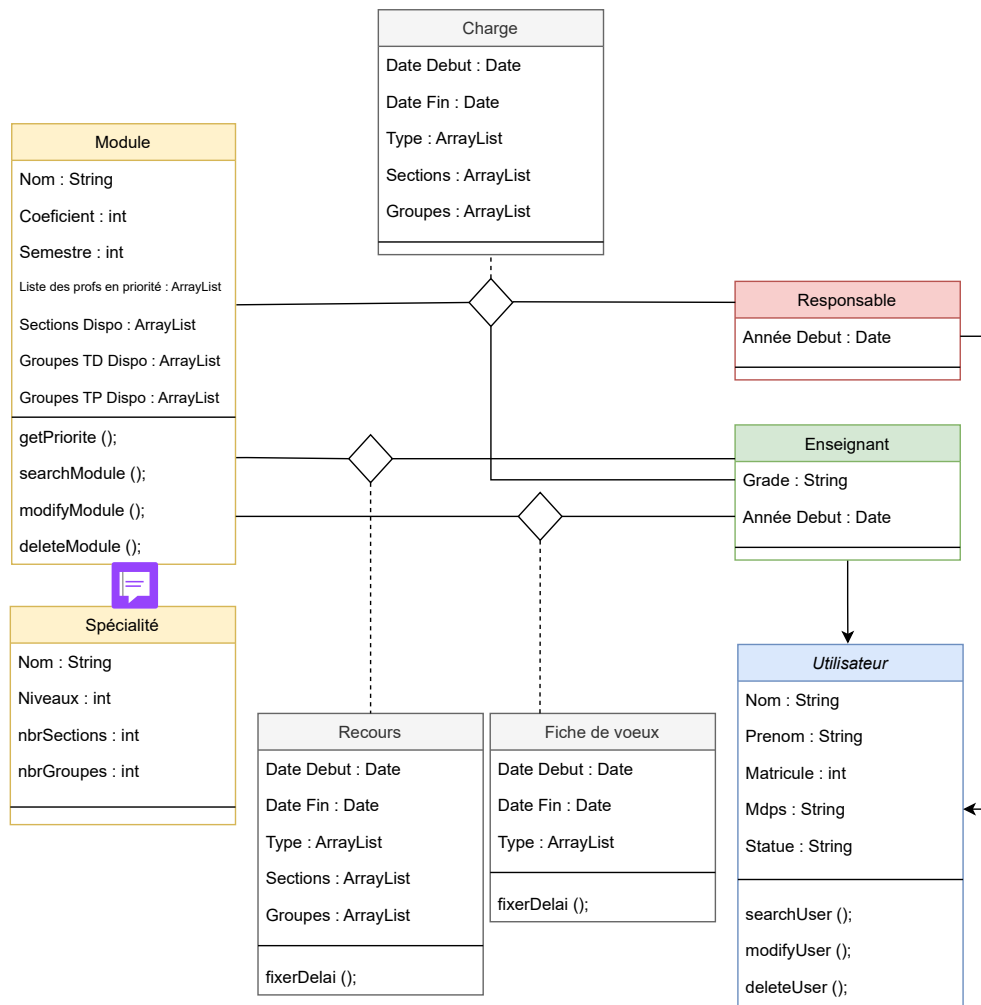



FIGURE 3.1 – Diagramme de classe

## 3.3 Passage au modèle relationnel

Dans cette ~~phase~~, nous allons créer le modèle relationnel de données en se basant sur le diagramme de classes afin de pouvoir implémenter notre base de données.

### 3.3.1 Règles de transformation du modèle objet vers le modèle relationnel :

Le passage depuis le modèle objets (D C L) vers le modèle relationnel  t dans l'objectif d'implémenter la base de données sur un S G BD relationnel.

-Pour passer d'une représentation orientée objet en D C L vers le modèle relationnel, il faut appliquer Les règles de passage suivantes.

- Une classe correspond à une table du modèle relationnel :
- Chaque attribut donne lieu à une colonne.
- Chaque instance stocke ses données dans une ligne et son (ID) sert de clé primaire.
- Certains types complexe ~~en~~ correspondent à aucun des types SQL ; on rencontre fréquemment ce cas pour les attributs représentant une structure de donnée.
- Un type complexe peut être conçu :
- Soit avec plusieurs colonnes, chacune correspondant à une structure.
- Soit avec une table spécifique dotée d'une clé étrangère pour relier les instances aux valeurs de leurs attributs complexes.

Pour transformer un diagramme de classe en modèle relationnel, il existe 3 règles à suivre : [6]

#### Règle 1 (Cardinalité (?.1)) :

Chaque classe est transformée en une table. Chaque attribut de classe est transformé en un champ de table. L'identifiant de la classe qui a une cardinalité (1) devient une clé étrangère dans l'autre classe.

#### Règle 2 (Cardinalité (?.N)) :






Chaque classe est transformée en une table. Chaque attribut de classe est transformé en un champ de table. L'association est transformée en une table distincte. Cette table contient les identifiants de chaque classe liée, ainsi que tout autre attribut pertinent.

#### Règle 3 (Présence d'une généralisation) :

- Méthode 1 : Création d'une seule table pour tous les attributs des classes impliquées dans la généralisation, avec un attribut supplémentaire pour distinguer les types d'objets.
- Méthode 2 : Création de tables distinctes pour chaque sous-type, chaque table contenant les attributs généraux ainsi que les attributs spécifiques à ce sous-type.

### 3.3.2 Règles de transformation du modèle objet vers le modèle relationnel :

Après avoir réalisé le diagramme de classe, ~~on~~ appliquant les règles de passage ~~du~~ modèle relationnel ~~et on va déterminer~~ le schéma de la base de données relationnelle qui permet de spécifier les différentes tables avec les différentes relations entre-elles :

- **Utilisateur** : (matricule, nom, prénom, mdps, statut, grade, année\_expérience)
- **Enseignant** : (#matricule, année\_début)
- **Responsable** : (#matricule, année\_début)
- **Spécialité** : (id\_specialite, nom, niveaux, nbr\_sections, nbr\_groupes)
- ~~Section~~ : (id\_section, niveau, ~~#id\_specialite~~)
- ~~Groupe~~ : (id\_groupe, type, ~~#id\_section~~)
- **Module** : (id\_module, nom, coefficient, semestre, ~~niveau~~, #id\_specialite)
- **Module\_Enseign**  : (#id\_module, #matricule, priorité)
- **FicheDeVoeux** : (id\_fiche  #matricule, date\_début, date\_fin, type)
- **Charge** : (id\_charge, #matricule  #id\_module, date\_début, date\_fin, type 
- ~~Charge\_Section~~ : (#id\_charge, ~~#id\_section~~)
- ~~Charge\_Groupe~~ : (#id\_charge, ~~#id\_groupe~~)
- **Recours** : (id\_recours, #matricule, #id\_module, date\_début, date\_fin, type 

— **Recours\_Section** : (#id\_recours, #id\_section)

— **Recours\_Groupe** : (#id\_recours, #id\_groupe)

**Remarque** : Les clés primaires sont soulignées, et les clés étrangères sont précédées par un(#).

### 3.4 Description des tables de la Base des données

Attribut	Type	Description	Contrainte
matricule	VARCHAR(20)	Identifiant unique de l'utilisateur	Primary key, Not null
nom	VARCHAR(70)	Nom de l'utilisateur	Not null
prénom	VARCHAR(70)	Prénom de l'utilisateur	Not null
mdps	VARCHAR(255)	Mot de passe (haché)	Not null
statut	VARCHAR(50)	Rôle (Enseignant, Responsable, etc.)	Not null
grade	VARCHAR(50)	Grade académique	
année_expérience	INT	Années d'expérience	

TABLE 3.1 – Table Utilisateur

Attribut	Type	Description	Contrainte
matricule	VARCHAR(20)	Référence à un utilisateur enseignant	Primary key, Foreign key
année_début	YEAR	Année de début en tant qu'enseignant	Not null

TABLE 3.2 – Table Enseignant

Attribut	Type	Description	Contrainte
matricule	VARCHAR(20)	Référence à un utilisateur responsable	Primary key, Foreign key
année_début	YEAR	Année de prise de responsabilité	Not null

TABLE 3.3 – Table Responsable

Attribut	Type	Description	Contrainte
id_specialite	INT	Identifiant de la spécialité	Primary key
nom	VARCHAR(100)	Nom de la spécialité	Not null
niveaux	VARCHAR(50)	Niveaux associés (ex : L1, M2...)	
nbr_sections	INT	Nombre de sections	
nbr_groupes	INT	Nombre de groupes	

TABLE 3.4 – Table Spécialité

Attribut	Type	Description	Contrainte
id_section	INT	Identifiant unique de la section	Primary key
niveau	VARCHAR(10)	Niveau associé (ex : L3, M1...)	Not null
id_specialite	INT	Référence à la spécialité	Foreign key, Not null

~~TABLE 3.5 – Table Section~~

Attribut	Type	Description	Contrainte
id_groupe	INT	Identifiant unique du groupe	Primary key
type	VARCHAR(30)	Type de groupe (TP, TD, etc.)	Not null
id_section	INT	Référence à la section	Foreign key, Not null

~~TABLE 3.6 – Table Groupe~~

Attribut	Type	Description	Contrainte
id_module	INT	Identifiant du module	Primary key
nom	VARCHAR(100)	Nom du module	Not null
coefficient	FLOAT	Coefficient du module	Not null
semestre	VARCHAR(10)	Semestre associé	Not null
<del>niveau</del>	<del>VARCHAR(10)</del>	<del>Niveau concerné</del>	
id_specialite	INT	Référence à la spécialité	Foreign key, Not null

TABLE 3.7 – Table Module

Attribut	Type	Description	Contrainte
id_module	INT	Référence au module	Primary key, Foreign key
matricule	VARCHAR(20)	Référence à l'enseignant	Primary key, Foreign key
priorité	INT	Niveau de préférence du module	Not null

TABLE 3.8 – Table Module\_Enseignant


Attribut	Type	Description	Contrainte
id_fiche	INT	Identifiant de la fiche de vœux	Primary key
matricule	VARCHAR(20)	Référence à l'enseignant	Foreign key, Not null
date_début	DATE 	Date de début de la saisie	Not null
date_fin	DATE	Date de fin de la saisie	Not null
type	VARCHAR(30)	Type de fiche ( <del>prévisionnelle, définitive, etc.</del> )	Not null

TABLE 3.9 – Table FicheDeVoeux

Attribut	Type	Description	Contrainte
id_charge	INT	Identifiant de la charge	Primary key
matricule	VARCHAR(20)	Référence à l'enseignant	Foreign key, Not null
id_module	INT	Référence au module	Foreign key, Not null
date_début	DATE	Date de début de la charge	Not null
date_fin	DATE	Date de fin de la charge	Not null
type	VARCHAR(30)	Type de charge (TP, TD, C, etc.)	Not null

TABLE 3.10 – Table Charge

Attribut	Type	Description	Contrainte
id_charge	INT	Référence à la charge	Primary key, Foreign key
id_section	INT	Référence à la section	Primary key, Foreign key

~~TABLE 3.11 – Table Charge\_Section~~

Attribut	Type	Description	Contrainte
id_charge	INT	Référence à la charge	Primary key, Foreign key
id_groupe	INT	Référence au groupe	Primary key, Foreign key

~~TABLE 3.12 – Table Charge\_Groupe~~

Attribut	Type	Description	Contrainte
id_recours	INT	Identifiant du recours	Primary key
matricule	VARCHAR(20)	Référence à l'enseignant	Foreign key, Not null
id_module	INT	Référence au module	Foreign key, Not null
date_début	DATE	Date de début du recours	Not null
date_fin	DATE	Date de fin du recours	Not null
type	VARCHAR(30)	Type de recours (TP, TD, C, etc.)	Not null

TABLE 3.13 – Table Recours

### 3.5 Conclusion

Dans ce chapitre, nous avons réalisé la modélisation de la structure statique de notre système à l'aide du diagramme de classes. Cette modélisation nous a permis de concevoir la base de données sous-jacente et de développer notre application selon une approche orientée objet. Dans le prochain chapitre, nous présenterons en détail l'application et ses différents éléments, offrant ainsi un aperçu complet de notre projet.



# **Chapitre 4**

## **Implémentation**

### **4.1 Introduction**

### **4.2 Environnement de travail**

### **4.3 Interfaces graphiques**

# **Bibliographie**