

RAPPORT DE CONCEPTION

Projet : Une plateforme de gestion des projets « un diplôme-une startup / un diplôme-un brevet ».



Devnetly

Table des versions

Version	Date De Sortie	Auteur	Changement
1.0	25 Mars 2023	Benyamina Yacine Lazreg	Première version
1.1	18 Avril 2023	Benyamina Yacine Lazreg	Deuxième version
1.2	08 Mai 2023	Benyamina Yacine Lazreg	Troisième version
1.3	01 Juin 2023	Benyamina Yacine Lazreg	Quatrième version

TABLE DES MATIÈRES

1	Introduction	3
2	Les Outils de Modélisation	4
2.1	Astah	4
2.2	MySQL Workbench	4
3	Premier Incrément : Gestion des comptes et des rôles	5
3.1	Les Diagrammes Conceptuels	5
3.1.1	Diagramme de Classe	5
3.1.2	Le Diagramme De Relation d'Entité Amélioré	7
3.2	Modèle Relationnelle	7
4	Deuxième Incrément : Gestion des projets	9
4.1	Les Diagrammes Conceptuels	10
4.2	Modèle Relationnelle	11
5	Troisième Incrément : Suivi des projets	13
5.1	Les Diagrammes Conceptuels	14
5.2	Modèle Relationnelle	15
6	Conclusion	17

CHAPITRE

1

INTRODUCTION

La conception d'un projet est une étape cruciale car son but premier est de permettre de créer un système ou un processus répondant aux besoins en tenant compte des contraintes. Le système en question doit être suffisamment défini pour pouvoir être installé, fabriqué, construit et être fonctionnel, et pour répondre aux besoins du client. À travers ce rapport, la conception de notre projet sera entamée en utilisant l'outil UML dans l'intention de préparer son implémentation et rendre le travail des développeurs plus précis. On verra dans ce qui suit : les outils utilisés pour réaliser notre conception, la réalisation du diagramme de classes de conception des trois premiers incréments, le modèle Conceptuel de données et modèle relationnel associé.

CHAPITRE

2

LES OUTILS DE MODÉLISATION

2.1 Astah

Astah UML est un outil de conception logicielle qui permet aux utilisateurs de créer des diagrammes UML (Unified Modeling Language) pour des projets de développement logiciel. Avec Astah UML, les utilisateurs peuvent créer différents types de diagrammes UML, y compris des diagrammes de classe, des diagrammes de cas d'utilisation, des diagrammes de séquence, etc. Astah UML est conçu pour aider les développeurs à visualiser et à concevoir des projets logiciels de manière claire et efficace.

2.2 MySQL Workbench

MySQL Workbench est un outil de modélisation de données et d'administration de base de données pour MySQL. Il permet de concevoir, de modéliser, de générer et de gérer des bases de données MySQL visuellement. Il comprend des fonctionnalités telles que la conception de schémas de base de données, la modélisation entité-association (ER), la création de tables et d'objets de base de données, la modification de données, la sauvegarde et la restauration de bases de données, la gestion des utilisateurs et des privilèges, et bien plus encore. MySQL Workbench est une solution tout-en-un pour les développeurs de bases de données qui utilisent MySQL.

CHAPITRE

3

PREMIER INCRÉMENT: GESTION DES COMPTES ET DES RÔLES

3.1 Les Diagrammes Conceptuels

Un schéma conceptuel de données donne une visualisation graphique de la structure conceptuelle d'un système d'information et facilite l'identification des principales entités à représenter, de leurs attributs et des relations entre elles. Les schémas ci-dessous concernent le premier incrément : "Gestion des comptes et des rôles".

3.1.1 Diagramme de Classe

Un diagramme de classe est un type de diagramme utilisé en génie logiciel pour visualiser la structure des classes d'un système et leurs relations. Les classes sont des modèles qui représentent des entités réelles ou imaginaires, telles que des objets, des personnes, des concepts ou des transactions, et qui définissent les propriétés et les comportements de ces entités. Le diagramme de classe représente ces classes sous forme de boîtes avec des attributs et des méthodes, et utilise des flèches pour montrer les relations entre les classes, telles que l'héritage, l'association, l'agrégation et la composition. Les diagrammes de classe sont utiles pour planifier la conception d'un système orienté objet et pour communiquer les concepts de conception aux développeurs et aux autres membres de l'équipe de projet.

Ci-dessous la Figure 1 représentant notre diagramme de classes de conception du premier incrément :

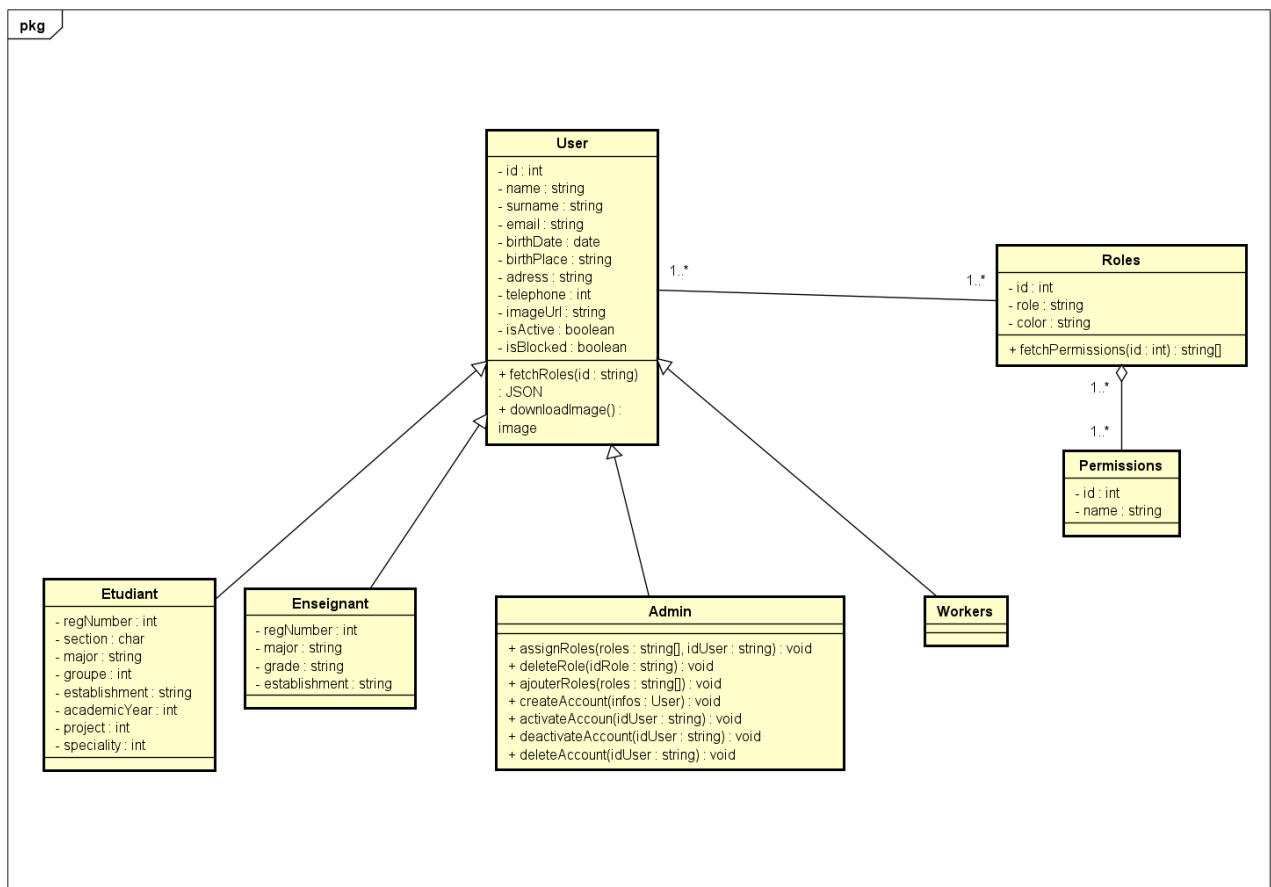


FIGURE 3.1 – Diagramme de Classe de Conception de l'incrément "Gestion des comptes et des rôles"

3.1.2 Le Diagramme De Relation d'Entité Amélioré

Le diagramme de relation d'entité amélioré (ERD amélioré) est une extension du modèle entité-association (ER) utilisé pour la conception de bases de données. Il permet de représenter les relations entre les entités en utilisant des symboles spécifiques pour indiquer la cardinalité des relations, la participation des entités dans ces relations, ainsi que les attributs associés à ces relations. En plus des entités et des relations, l'ERD amélioré comprend également des concepts tels que les domaines d'attributs, les clés primaires et étrangères, les index et les contraintes d'intégrité.

L'ERD amélioré est utile pour la modélisation de bases de données complexes en permettant une représentation visuelle claire de la structure et des relations des données. Il est couramment utilisé pour la conception de systèmes de gestion de bases de données relationnelles.

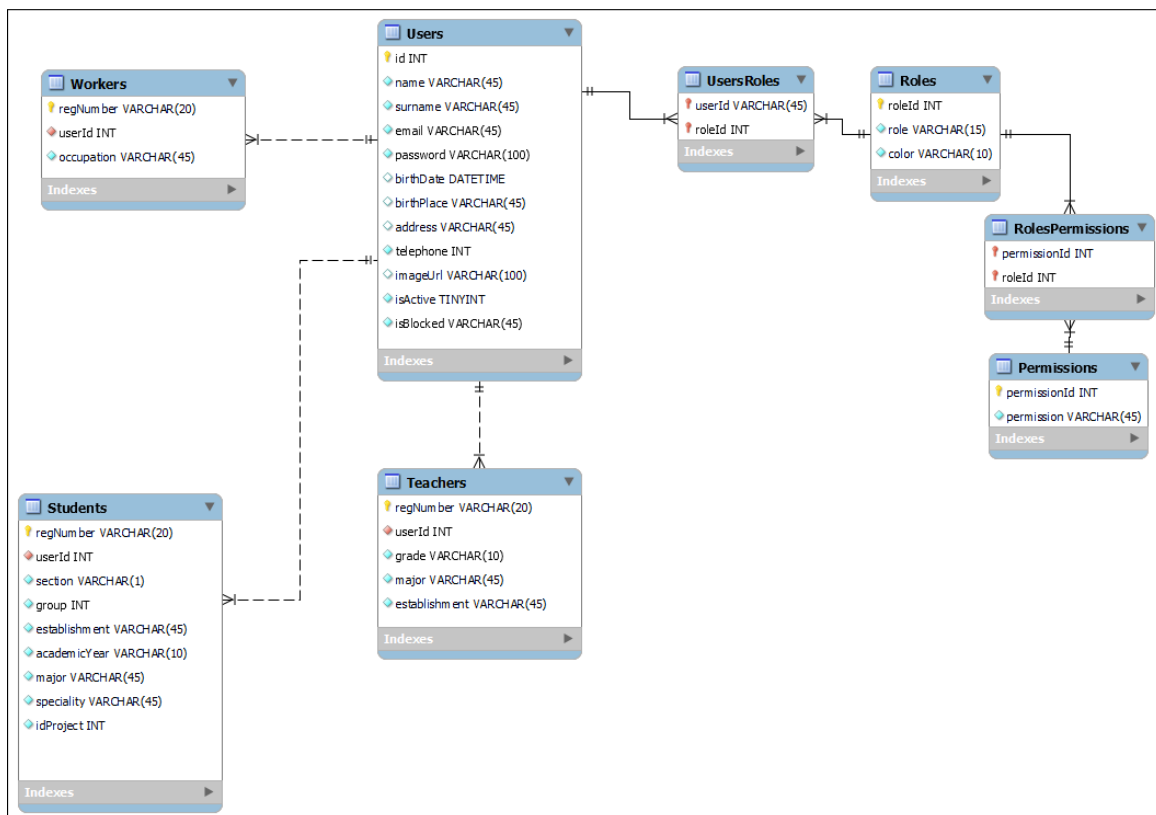


FIGURE 3.2 – ERD amélioré de l'incrément "Gestion des comptes et des rôles"

Les contraintes

- Les identifiants des étudiants et des professeurs sont les mêmes que leurs identifiants dans la plateforme "Progress", les identifiants des admins et service de stage sont les mêmes identifiants qui leur ont été attribués par l'établissement.

3.2 Modèle Relationnelle

Le modèle relationnel est un modèle de données qui décrit la structure des données d'un système d'information sous forme de tables. Chaque table représente une entité du système et chaque ligne de la table représente une instance de cette entité. Les relations entre les entités

sont établies en créant des clés étrangères dans les tables correspondantes. Les clés étrangères permettent de relier les données d'une table à celles d'une autre table. Il est utilisé pour la conception de bases de données relationnelles, qui sont largement utilisées dans les applications d'entreprise et les systèmes d'information. Les bases de données relationnelles permettent de stocker, de manipuler et de récupérer des données de manière efficace et sécurisée.

C'est un résultat obtenu à travers le Modèle entité-association permettant de créer les tables utilisées dans notre futur base de données.

Table Roles :

Roles(roleId, role, color)

Table Permissions :

Permissions(permissionId, permission)

Table RolesPermissions :

Permissions_Roles(#permissionId, #roleId)

Table Users :

Users(id, name, surname, email, password, birthDate, birthPlace, adress, telephone, imageUrl, isActive, isBlocked)

Table Workers :

Workers(#userId, regNumber, occupation)

Table Students :

Students(#userId, regNumber, section, major, group, establishment, academicYear, speciality)

Table Teachers :

Teachers(#userId, regNumber, major, establishment, grade)

Table UsersRoles :

UsersRoles(#userId, #roleId)

CHAPITRE

4

DEUXIÈME INCRÉMENT: GESTION DES
PROJETS

4.1 Les Diagrammes Conceptuels

Diagramme de Classes de Conception

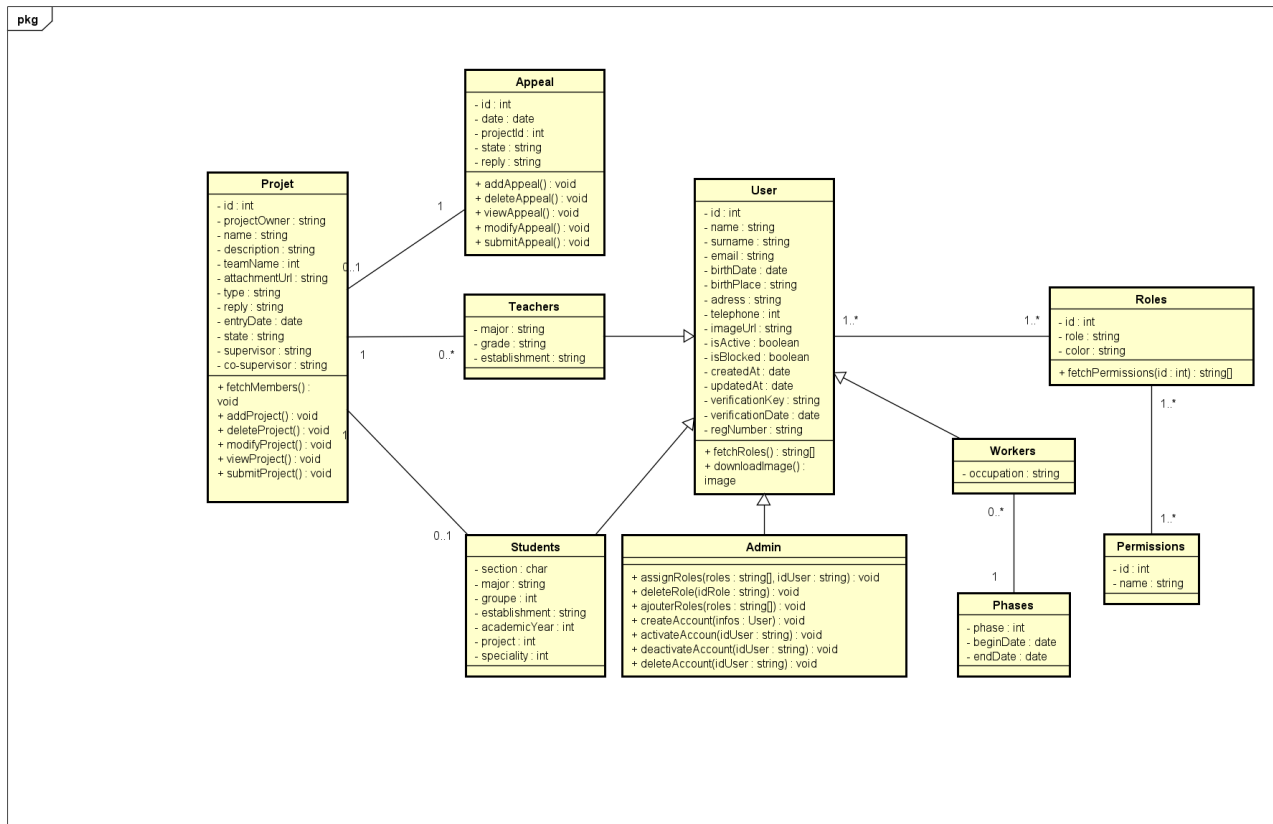


FIGURE 4.1 – Diagramme de Classe de Conception de l'incrément "Gestion des projets"

Le Diagramme De Relation d'Entité Amélioré

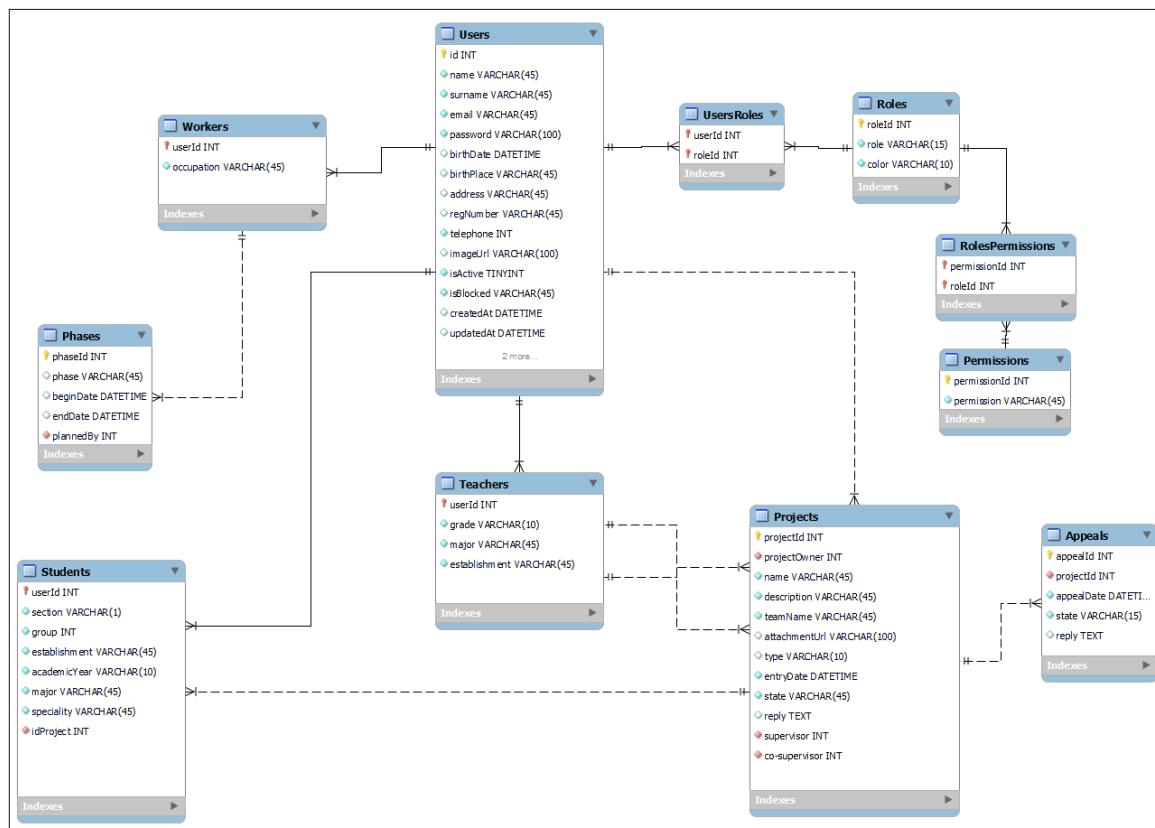


FIGURE 4.2 – ERD amélioré de l'incrément "Gestion des projets"

4.2 Modèle Relationnelle

Table Roles :

Roles(roleId, role, color)

Table Permissions :

Permissions(permissionId, permission)

Table RolesPermissions :

RolesPermissions(#permissionId, #roleId)

Table Users :

Users(id, name, surname, email, password, regNumber, birthDate, birthPlace, address, telephone, imageUrl, isActive, isBlocked, createdAt, updatedAt, verificationKey, verificationDate)

Table Workers :

Workers(#userId, occupation)

Table Students :

Students(#userId, section, major, group, establishment, academicYear, speciality, projectId)

Table Teachers :

Teachers(#userId, major, establishment, grade)

Table UsersRoles :

UsersRoles(#userId, #roleId)

Table Projects :

Projects(projectId, #projectOwner, name, description, attachmentUrl, teamName, entryDate, state, reply, type, #supervisor, #co-supervisor)

Table Appeals :

Appeals(appealId, #projectId, appealDate, state, reply)

Table Phases :

Phases(phaseId, phase, beginDate, beginDate, #plannedBy)

CHAPITRE

5

TROISIÈME INCRÉMENT: SUIVI DES
PROJETS

5.1 Les Diagrammes Conceptuels

Diagramme de Classes de Conception

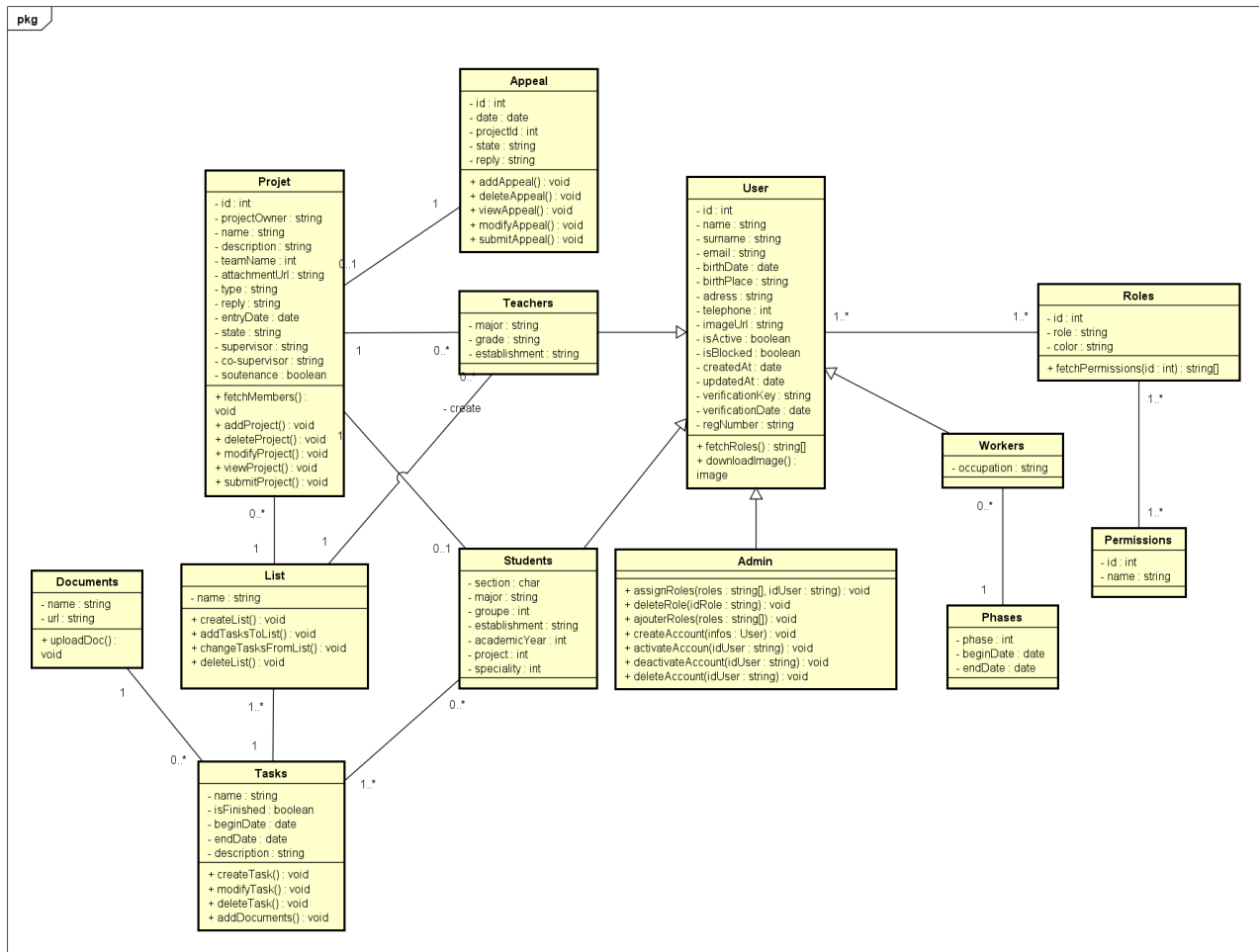


FIGURE 5.1 – Diagramme de Classe de Conception de l'incrément "Suivi des projets"

Le Diagramme De Relation d'Entité Amélioré

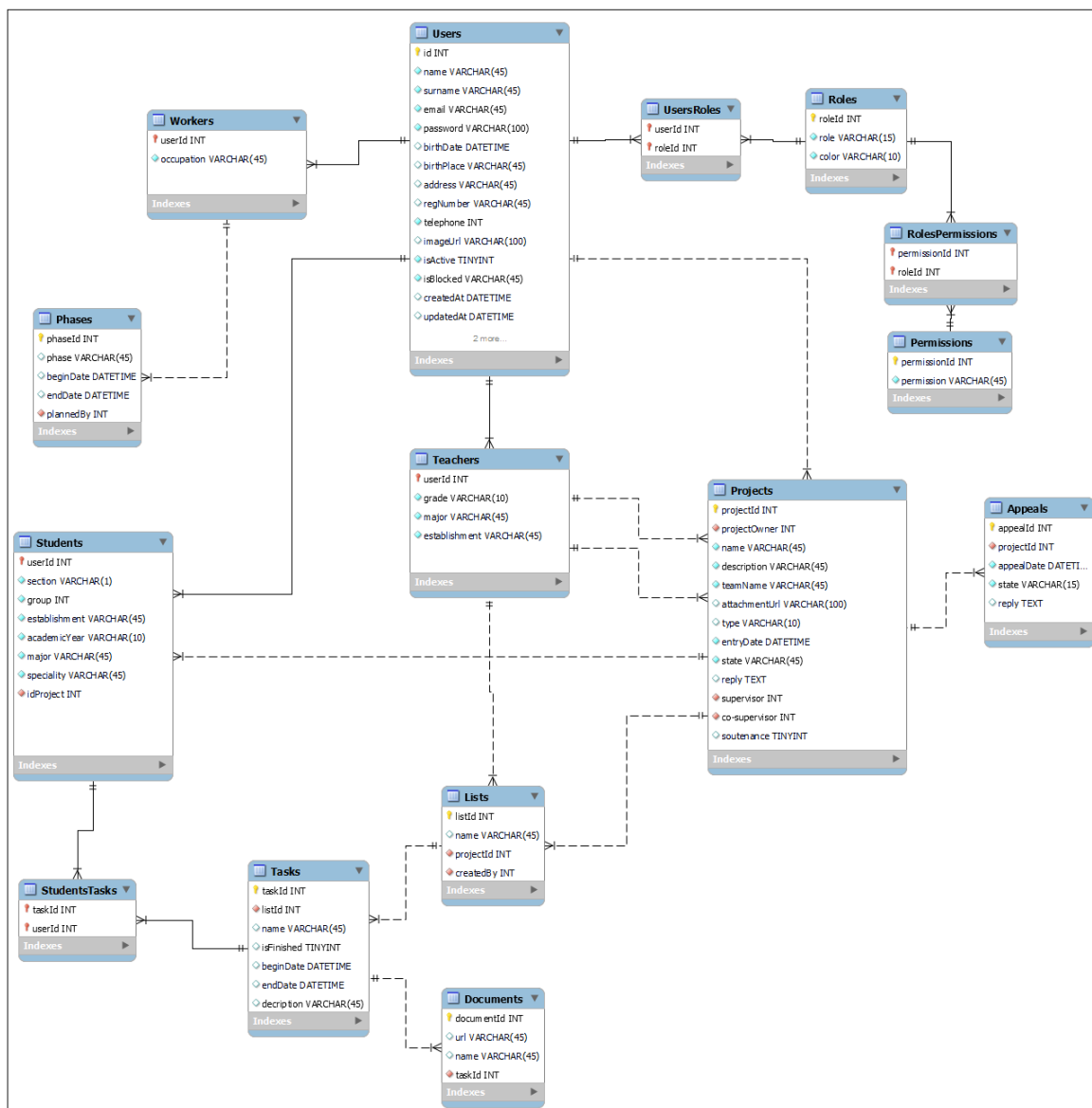


FIGURE 5.2 – ERD amélioré de l'incrément "Suivi des projets"

5.2 Modèle Relationnelle

Table Roles :

Roles(roleId, role, color)

Table Permissions :

Permissions(permissionId, permission)

Table RolesPermissions :

RolesPermissions(#permissionId, #roleId)

Table Users :

Users(id, name, surname, email, password, regNumber, birthDate, birthPlace, address, telephone, imageUrl, isActive, isBlocked, createdAt, updatedAt, verificationKey, verificationDate)

Table Workers :

Workers(#userId, occupation)

Table Students :

Students(#userId, section, major, group, establishment, academicYear, speciality, projectId)

Table Teachers :

Teachers(#userId, major, establishment, grade)

Table UsersRoles :

UsersRoles(#userId, #roleId)

Table Projects :

Projects(projectId, #projectOwner, name, description, attachmentUrl, teamName, entryDate, state, reply, type, soutenance, #supervisor, #co-supervisor)

Table Appeals :

Appeals(appealId, #projectId, appealDate, state, reply)

Table Phases :

Phases(phaseId, phase, beginDate, endDate, #plannedBy)

Table Lists :

Lists(listId, name, #createdBy, #projectId)

Table Tasks :

Tasks(taskId, name, isFinished, beginDate, endDate, description, #listId)

Table StudentsTasks :

StudentsTasks(#taskId, #userId)

Table Documents :

Documents(documentId, name, url, #taskId)

CHAPITRE

6

CONCLUSION

Ce document de conception des trois incréments « Gestion des Comptes et des rôles », « Gestion des Projets » et « Suivi des Projets » ont permis d’avoir une vision plus détaillée de ces derniers en utilisant le diagramme de classe de conception, le modèle conceptuel de données ainsi que le modèle relationnel qui a offert une étude plus approfondie du système et cela pour faciliter le travail des développeurs.