

|  |
| --- |
| Projet base de données |
| Hugues Bégeot  Célia Bunouf  L3 Int – Groupe 4 |

Une image contenant objet

Description générée automatiquement

Table des matières

[Introduction 3](#_Toc8510380)

[Dictionnaire de données 4](#_Toc8510381)

[MCD [Modèle conceptuel de données] : 6](#_Toc8510382)

[MLD [Modèle Logique de Données] : 7](#_Toc8510383)

[Scénario d’exécution : 8](#_Toc8510384)

[Difficultés rencontrées : 10](#_Toc8510385)

[Points faibles : 11](#_Toc8510386)

[Points forts : 11](#_Toc8510387)

# Introduction

Dans le cours de Base de données, nous avions pour projet de produire un programme capable d’effectuer la gestion administrative d’une école avec ses élèves, professeurs, cours et groupes, ainsi que de s’occuper de la gestion des notes des étudiants.

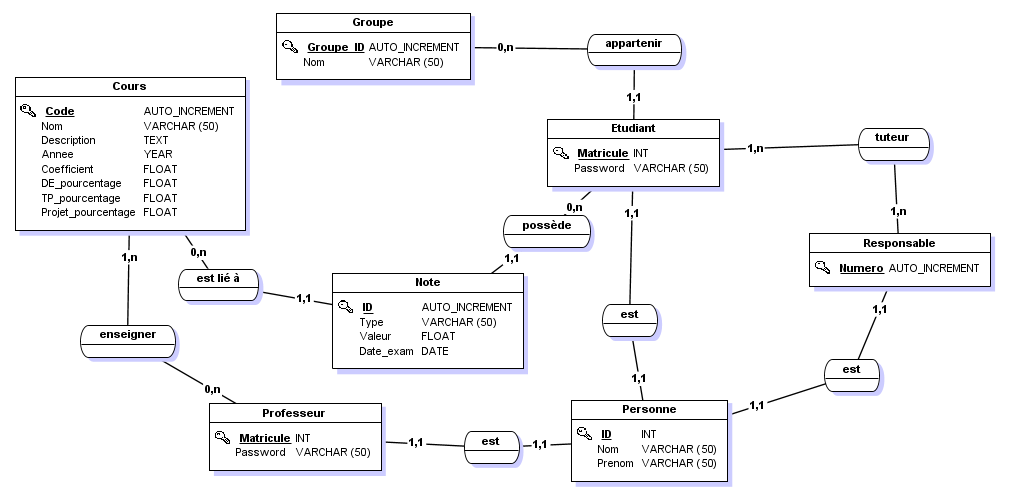
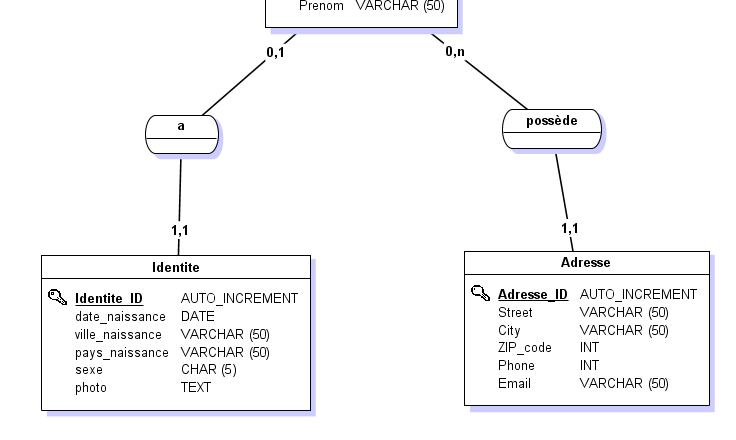
L’objectif final du projet est de créer une interface graphique permettant toutes les fonctionnalités décrites précédemment.

Le projet a été réalisé en **JAVA**, avec une base de données **MySQL**. L’interface a été réalisée à l’aide de **JavaSwing**.

# Dictionnaire de données

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Nom** | **Description** | **Domaine ou type** | **Taille** | **Commentaire** | **Contraintes,**  **règles de calcul** |
| Cours | Code | Code du cours | Auto\_increment |  |  | Automatique |
| Nom | Nom du cours | Varchar | 50 |  | Obligatoire |
| Description | Description du cours | Text |  |  |  |
| Annee | Année d'enseignement du cours | Year |  |  |  |
| Coefficient | coefficient du cours | Float |  |  |  |
| DE\_pourcentage | Pourcentage du DE associé au cours | Float |  |  |  |
| TP\_pourcentage | Pourcentage du TP associé au cours | Float |  |  |  |
| Projet\_pourcentage | Pourcentage du Projet | Float |  |  |  |
| Personne | ID | Index de la personne | Auto\_increment |  |  | Automatique |
| Nom | Nom du cours | Varchar | 50 |  | Obligatoire |
| Prenom | Prénom de la personne | Varchar | 50 |  |  |
| Adresse | Adresse\_ID | Identifiant de l'adresse | Auto\_increment |  |  | Automatique |
| Street | rue de l'adresse | Varchar | 50 |  |  |
| City | ville de l'adresse | Varchar | 50 |  |  |
| ZIP\_code | code postal de la ville | Int | 50 |  |  |
| Email | email de la personne | Varchar | 50 |  |  |
| Phone | numero de téléphone de la personne | Int |  |  |  |
| Identité | identite\_ID | Identifiant de l'identité | Auto\_increment |  |  | Automatique |
| date\_naissance | date de naissance | Date |  |  |  |
| ville\_naissance | ville de naissance | Varchar | 50 |  |  |
| pays\_naissance | pays de naissance | Varchar | 50 |  |  |
| sexe | sexe de la personne | Char | 5 |  |  |
| photo | photo de la personne | Text | 50 |  |  |
| Professeur/ Etudiant/ Administration | Matricule | Matricule du professeur/eleve | Int |  |  |  |
| Password | mot de passe du professeur/eleve | Varchar | 50 |  |  |
| Responsible | Numero | numero du responsible | Int |  |  | Automatique |
| Password | mot de passe de la personne | Varchar | 50 |  |  |
| Groupe | Nom | Nom du groupe | Boolean |  |  |  |
| Note | ID | Index de la personne | Auto\_increment |  |  | Automatique |
| Valeur | valeur de la note obtenue | Float |  |  |  |
| Type | type de note | Varchar | 50 | DE / TP ou Projet |  |
| Date\_exam | date de l'examen | Date |  |  |  |

# MCD [Modèle conceptuel de données] :

Lié aux limites de **JMerise** (version étudiante), toutes les classes n’ont pu être représentées sur le même MCD. De nombreuses classes et relations ont ainsi dû être codées manuellement.

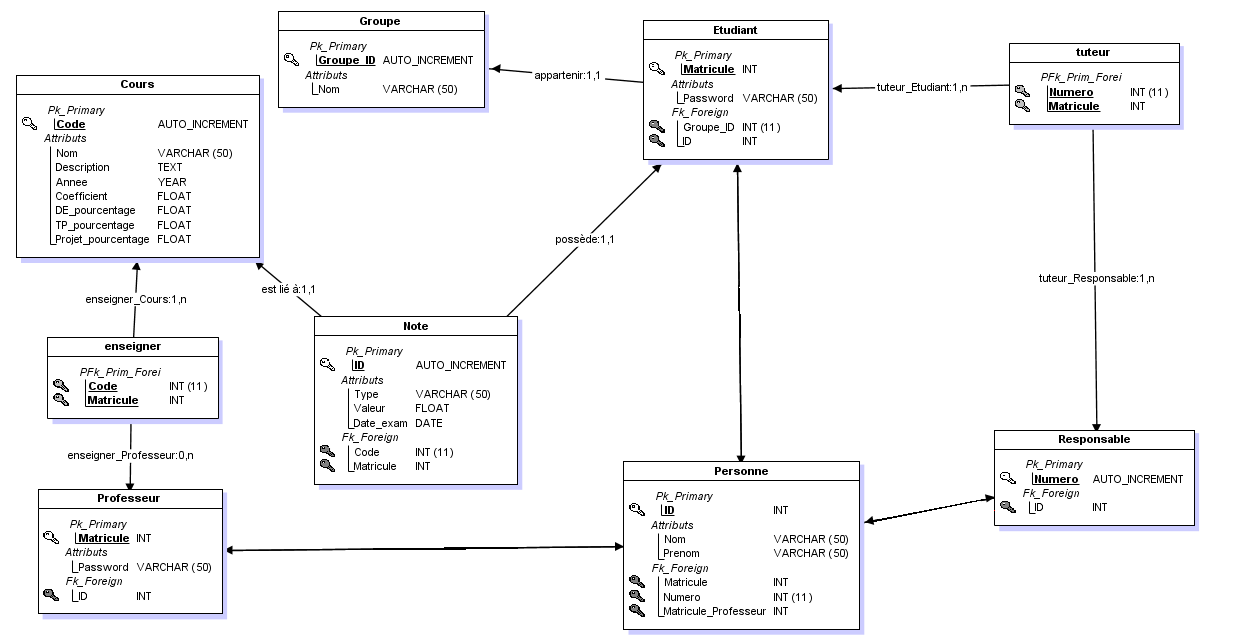
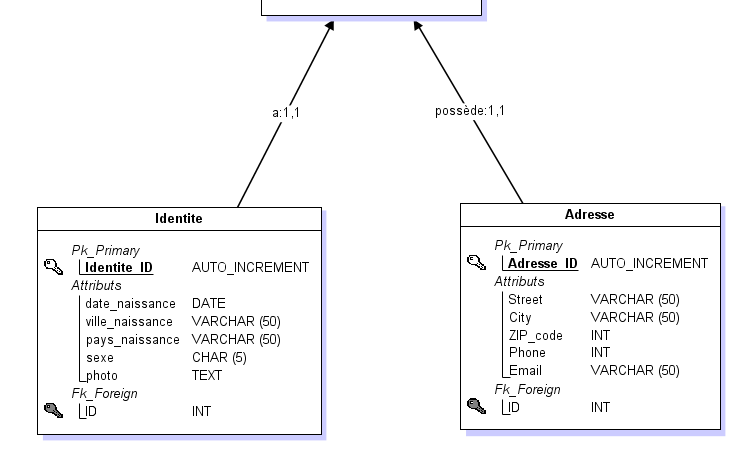
Nous avons ainsi regroupé plusieurs MCD pour tenter de représenter la base de données :

Nous avons donc décidé de partir sur un ensemble de 9 classes principales pour représenter l’ensemble d’une gestion d’école. Par la suite, une 10ième classe a été rajouté, la classe *administration* qui possède les mêmes caractéristiques qu’un *professeur* (matricule et password). Nous pensons que c’est une représentation plus que correcte de la réalité.

Par la suite, nous avons également dû ajouter un boolean bulletin dans la classe Groupe afin de représenter l’état de l’édition des bulletins.

# MLD [Modèle Logique de Données] :

Tout comme le MCD, le MLD n’a pu être réalisé entièrement automatiquement et à donc été implémenté manuellement.

 Voici une représentation MLD obtenu en combinant plusieurs MLD crées automatiquement :

Des tables ont également été modifiées manuellement.

# Scénario d’exécution :

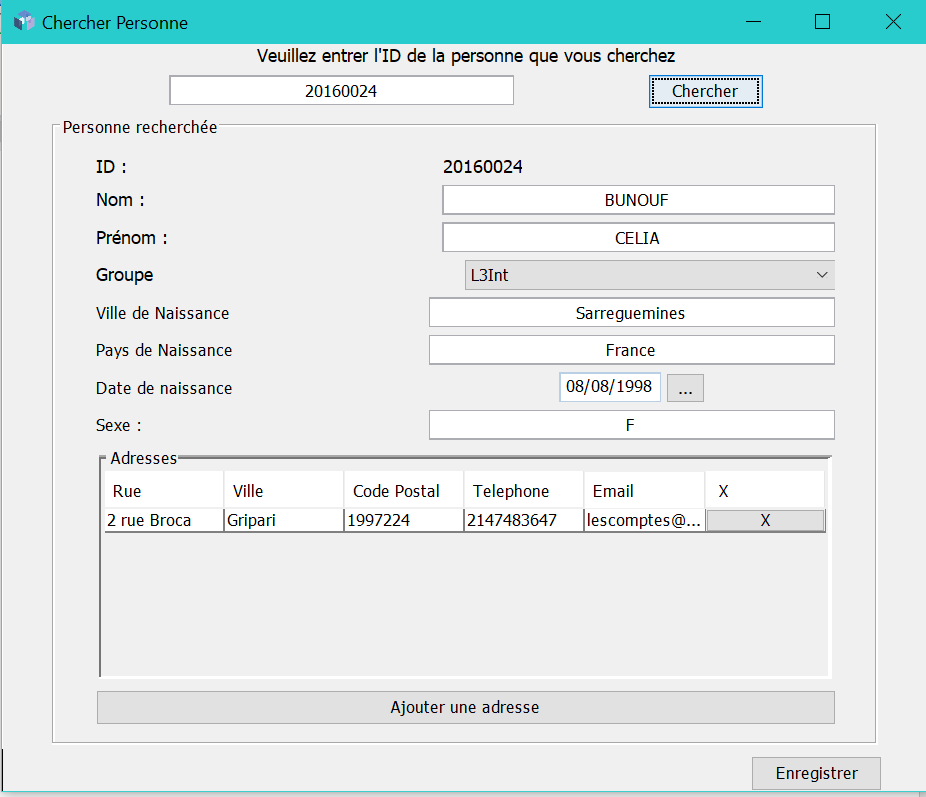
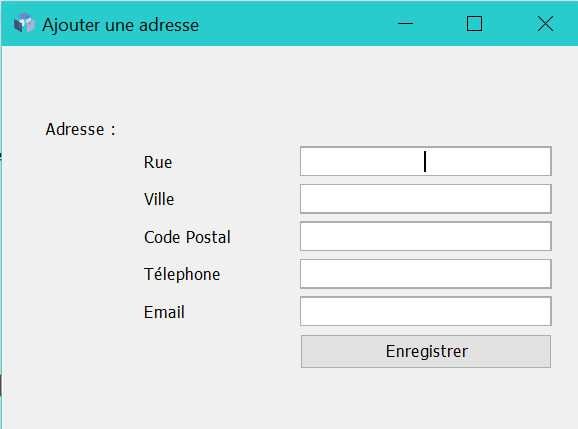
Tout d’abord, la personne utilisant l’application doit se connecter à l’aide de son matricule et de son mot de passe. Ainsi, chaque matricule est unique et un mot de passe lui est associé.

Ensuite, en fonction du statut de la personne dans l’établissement, différentes interfaces s’offrent à l’utilisateur. Ainsi, voici 3 cas différents :

## L’utilisateur fait partie de l’administration

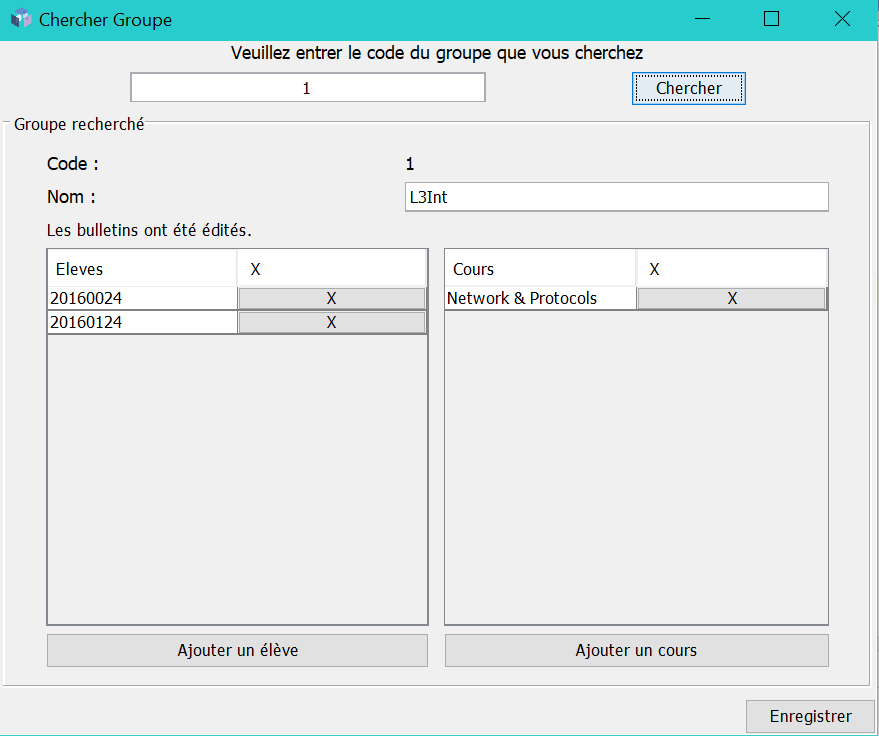
L’administration peut choisir ce qu’elle désire faire :

Si elle choisit de chercher un étudiant, le matricule de cet étudiant est demandé.

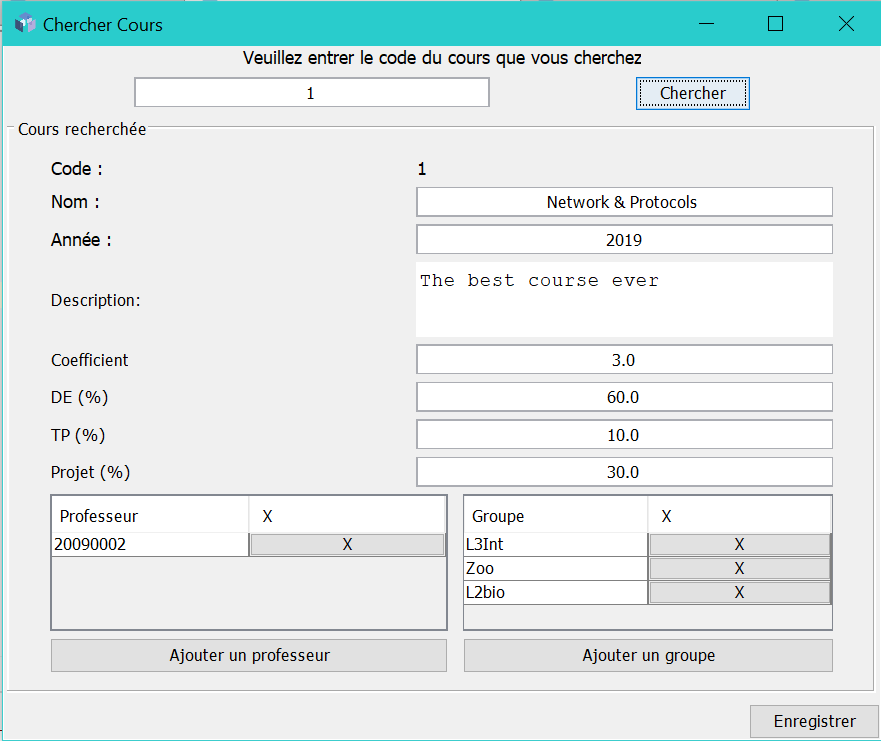
L’étudiant est affiché et il est possible de modifier les informations le concernant.

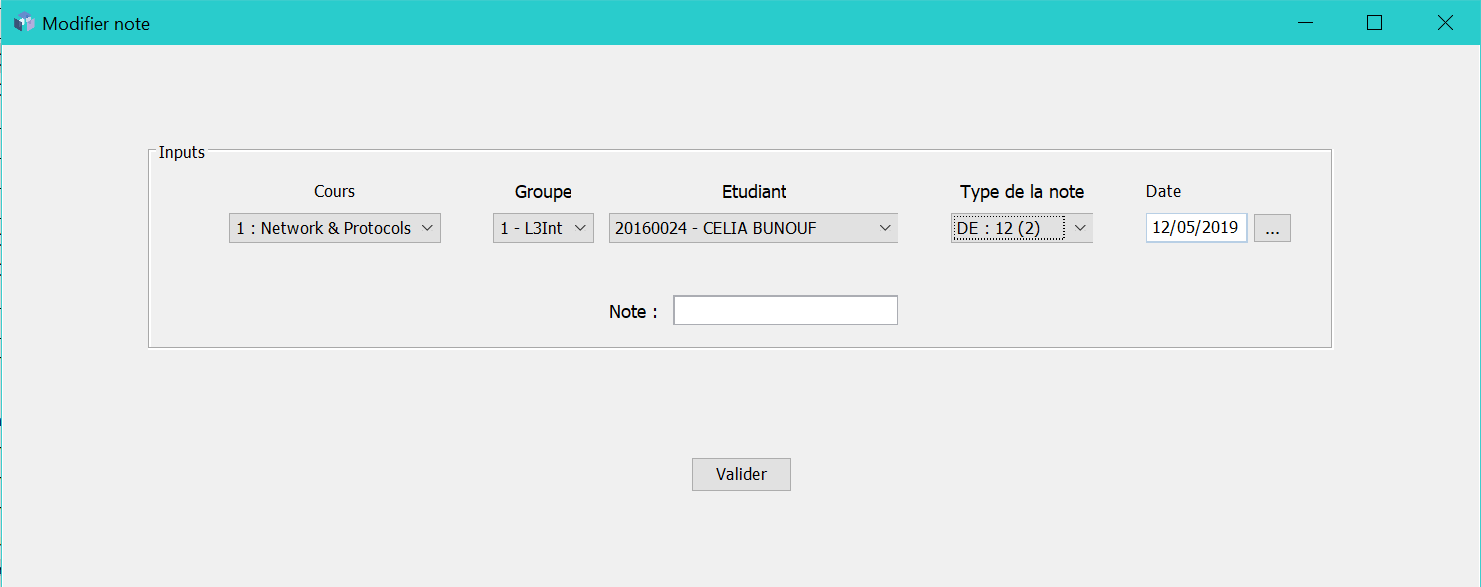
Il est ainsi possible d’ajouter ou supprimer les adresses de cet étudiant.

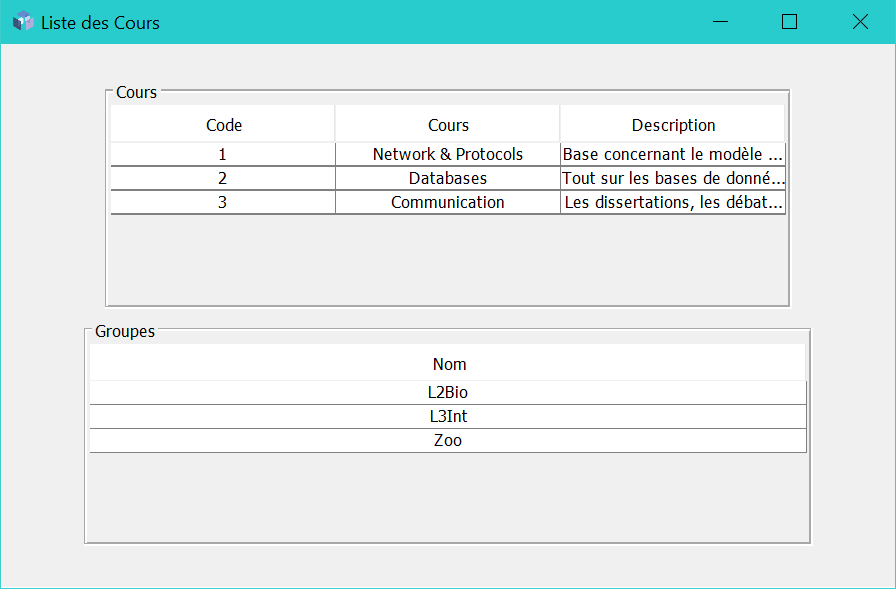
L’interface est sensiblement la même pour un professeur et un responsable.



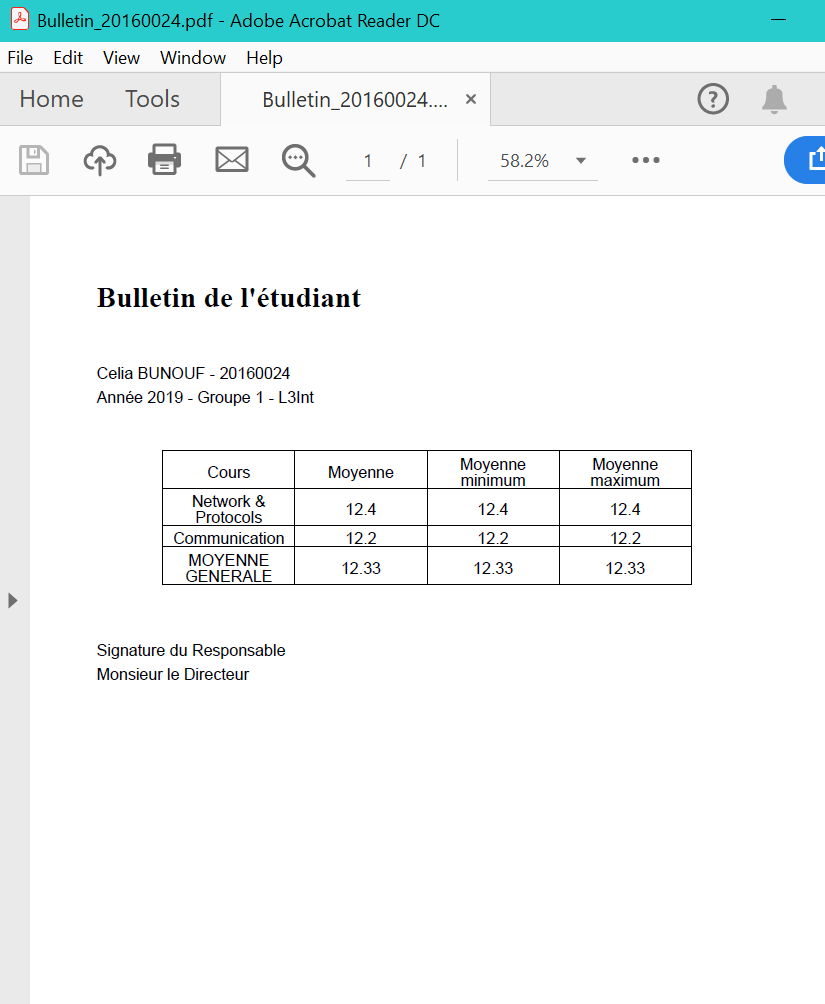
L’administration peut également chercher un groupe, ainsi qu’en modifier les cours suivis et les étudiants en faisant partie.

De plus, il est également possible de chercher un cours ainsi que modifier toutes les informations le concernant comme les professeurs qu’ils l’enseignent et les groupes qui le suivent.

L’administration peut également modifier des notes rentrées par les professeurs si ceux-ci leur demandent.

Ils peuvent également consulter les listes des cours et des groupes présents dans l’école.

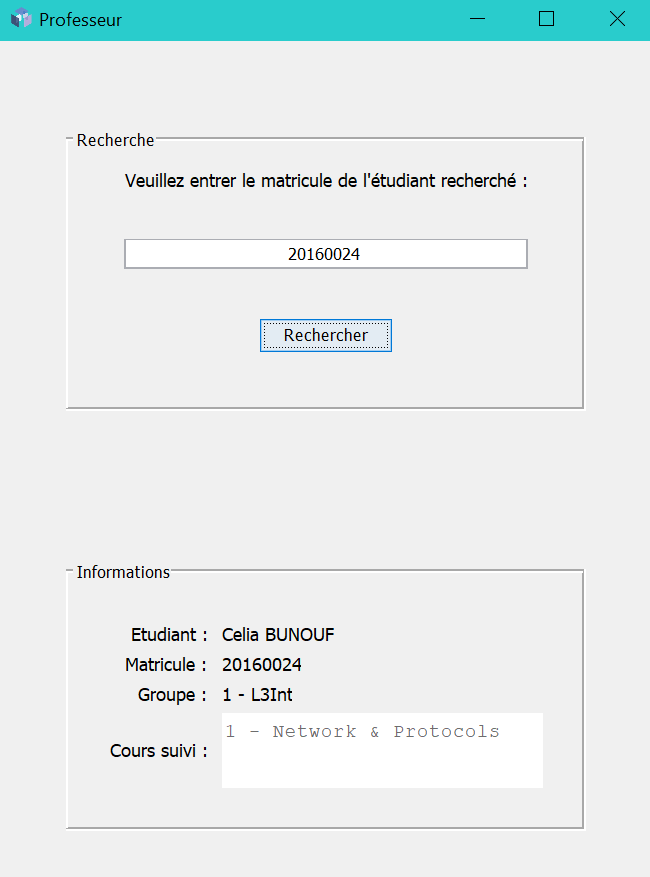
## L’utilisateur est un étudiant

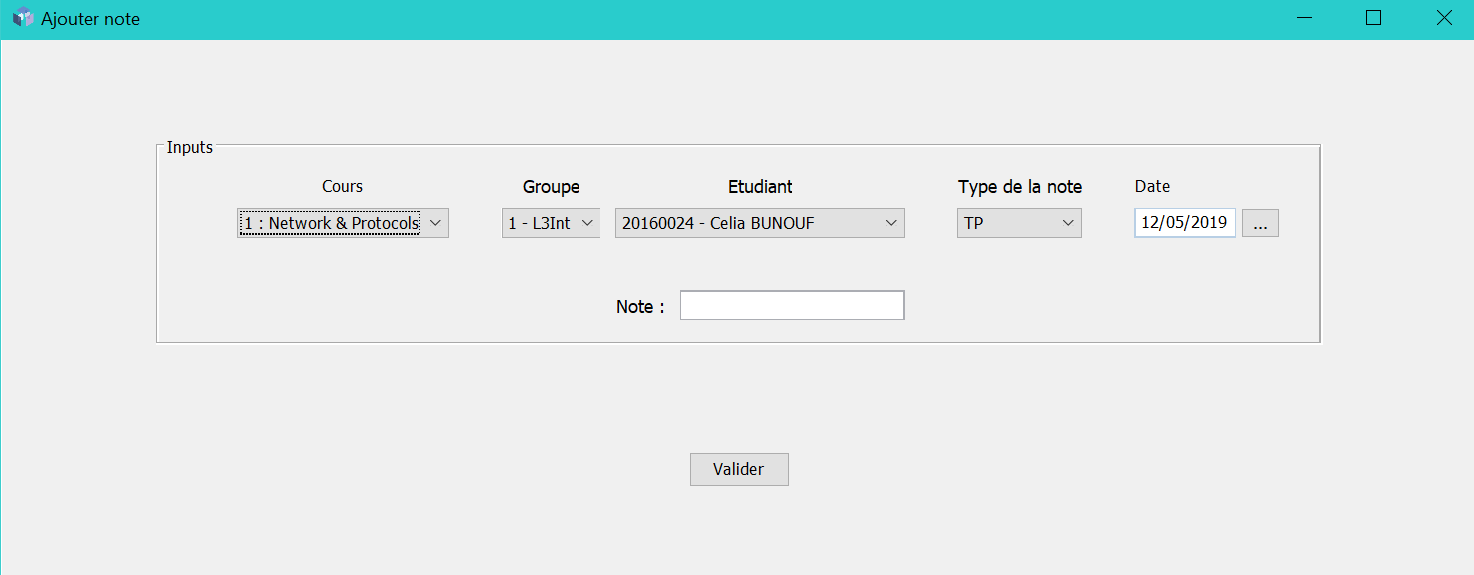
Dans le cas de l’étudiant, il peut consulter ses notes ainsi qu’imprimer son bulletin si celui-ci a été édité. Tout comme l’administration, l’étudiant peut consulter les listes de cours et de groupe.

Lorsque les bulletins ont été édités, l’étudiant peut l’enregistrer sur sa machine au format PDF, et l’imprimer par la suite par la suite s’il le souhaite.

## L’utilisateur est un professeur

Tout comme les étudiants et l’administration le professeur peut consulter les listes de cours et de groupe.

Il peut également chercher des étudiants.

De plus, il peut ajouter des notes aux étudiants.

# Difficultés rencontrées :

La difficulté rencontrée a été la connexion à la base de données au début car nous avions connecté notre IDE à la base de données et non le programme avant de comprendre qu’il suffisait d’inclure un fichier de connexion java.

Par la suite, on a vu que certains choix au niveau de la structure de la base de données n’étaient peut-être pas judicieux. Ils ont donc été modifié quand on a pu, et certains avait été oublié et ont donc été rajouté. On a ainsi pu se rendre compte que lors de la création d’une base de données on ne pense pas forcément à toutes les variables et environnement d’utilisation. Il est ainsi compliqué de créer la base de données idéales au programme dès le début de développement du projet.

# Points faibles :

Après implémentation de la base de données avec Java, nous avons remarqué que certains choix fait au départ n’étaient peut-être pas judicieux, notamment le fait d’avoir un ID pour le groupe car il n’est pas possible d’avoir deux groupes possédant le même nom, de ce fait, le nom doit être unique et non null et aurait pu être la seule clé primaire.

# Points forts :

Pour ce projet nous avons tenter de procurer une facilité d’utilisation et une plaisance visuelle de l’application. Nous pensons alors que l’un des points forts de notre projet est sa qualité d’interface, et sa facilité d’utilisation. Certaines modifications et fonctionnalités sont accessibles à différents endroits pour permettre une meilleure expérience utilisateur, comme par exemple le fait d’attribuer un professeur à un cours. Ceci peut être fait en recherchant un professeur ou en recherchant un cours. De même, pour savoir quel groupe suit quel cours, la modification peut être faite de la recherche du groupe ou de la recherche du cours.