



---

# CENTER CALL

---

Documentation AP4



LYCEE CLAUDE NOUGARO  
Lewis Crane, Eliaz Talmite, Basck Emmanuel

## Table des matières

1. Introduction .....	2
2. Cahier des charges (en résumé) .....	2
3. Modélisation des données.....	2
4. Applications et outils utilisés.....	2
5. Organisation du code.....	3
6. Manuel d'utilisation.....	3
7. Planning et gestion de projet.....	11
8. Problèmes rencontrés et solutions .....	11
9. Bilan et conclusion .....	11

# 1. Introduction

Dans le cadre de notre formation, nous avons réalisé une application back-office en client lourd Java Swing pour **Center Call**, permettant de gérer les utilisateurs. Elle vise à proposer des interfaces simples et fonctionnelles pour gérer les informations des utilisateurs.

## Objectifs :

- Développer une application installable sur poste local.
- Manipuler une base de données pour le stockage des données.
- Répondre au cahier des charges.
- Travailler en équipe.

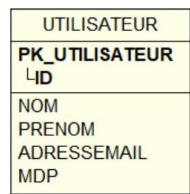
# 2. Cahier des charges (en résumé)

## Besoins fonctionnels :

- Avoir accès à la liste des utilisateurs.
- Ajouter, modifier, supprimer des utilisateurs.

# 3. Modélisation des données


## Modèle Conceptuel de Données :




## Explications :



La table **UTILISATEUR** sert à enregistrer les informations principales des utilisateurs, comme leur nom, prénom, adresse e-mail, identifiant et mot de passe. Chaque utilisateur est associé à un identifiant unique généré automatiquement.

# 4. Applications et outils utilisés

- **GitHub** : utilisé pour le versionnage du code source et la gestion collaborative du projet 

- **WampServer** : mis en place comme serveur local pour le développement et les phases de test. 

- **WinDesign** : employé pour la modélisation de la base de données. 

- **Google Drive** et **Microsoft Word** : utilisés pour la rédaction et le partage de la documentation.  

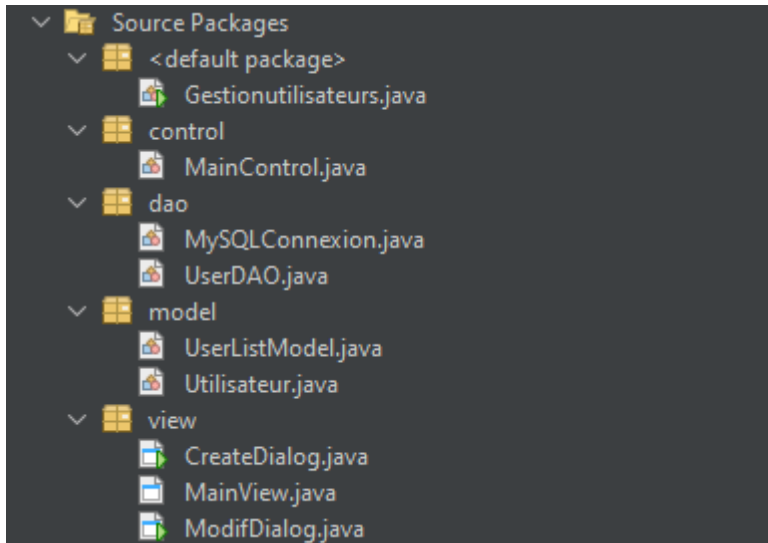


- **IDE NetBeans** : environnement de développement choisi pour le codage en Java.
- **Serveur Web LAMP** sur **Proxmox** : déployé pour héberger les données en environnement de préproduction.



## 5. Organisation du code

Nous avons choisi d'organiser notre code en plusieurs packages afin d'améliorer sa clarté et sa lisibilité.



**Le package View** : contient les classes de l'interface graphique réalisées avec Java Swing (fenêtres, dialogues, formulaires).

**Le package Model** : contient les classes métier correspondant aux entités définies à partir du MCD (Modèle Conceptuel de Données).

**Le package Control** : contient les classes de contrôle qui assurent la gestion des actions utilisateur et la liaison entre la vue et le modèle.

**Le package DAO (Data Access Object)** : contient les classes responsables de l'accès aux données (connexion à la base, requêtes SQL, opérations CRUD).

## 6. Manuel d'utilisation

**Installation :**

- Pour accéder au back-Office il suffit de cliquer sur le fichier « AP4.jar ».

<input type="checkbox"/> Nom	Modifié le	Type	Taille
archive-tmp	26/04/2025 18:30	Dossier de fichiers	
classes	26/04/2025 18:30	Dossier de fichiers	
generated-sources	26/04/2025 18:30	Dossier de fichiers	
generated-test-sources	26/04/2025 18:30	Dossier de fichiers	
maven-archiver	26/04/2025 18:30	Dossier de fichiers	
maven-status	26/04/2025 18:30	Dossier de fichiers	
test-classes	26/04/2025 18:30	Dossier de fichiers	
<input checked="" type="checkbox"/> AP4.jar	26/04/2025 18:30	Executable Jar File	

- Ensuite l'application va s'ouvrir.

Affichage
AP4 > target

Nom	Prenom	Email
King	Arthur	King@gmail.com
Choco	Nutella	Nutella@gmail.com
Electro	Darty	Darty@gmail.com
Doe	John	john.doe@example.com
Smith	Anna	anna.smith@example.c...
Martin	Lucas	lucas.martin@example...
Brown	Charlie	charlie.brown@exampl...
Taylor	Sophie	sophie.taylor@example...
Lemoine	Paul	paul.lemoine@example...
Durand	Emma	emma.durand@exampl...
Petit	Louis	louis.petit@example.com

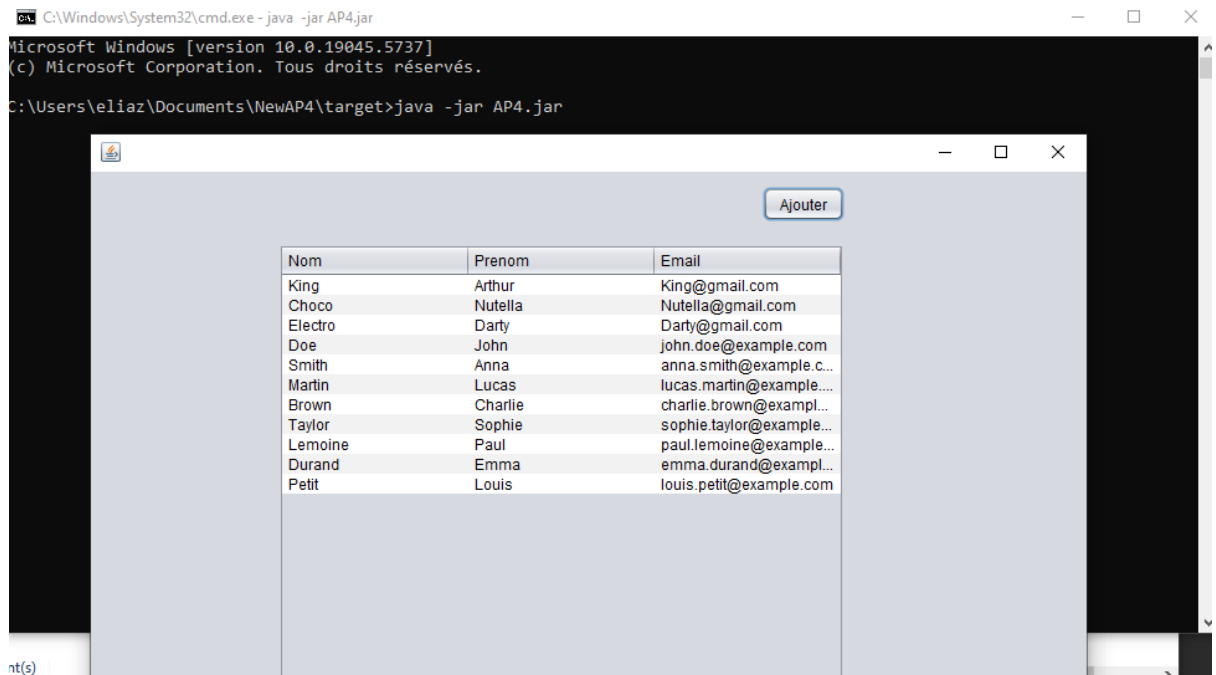
sectionné 4,29 Mo

modifier

supprimer

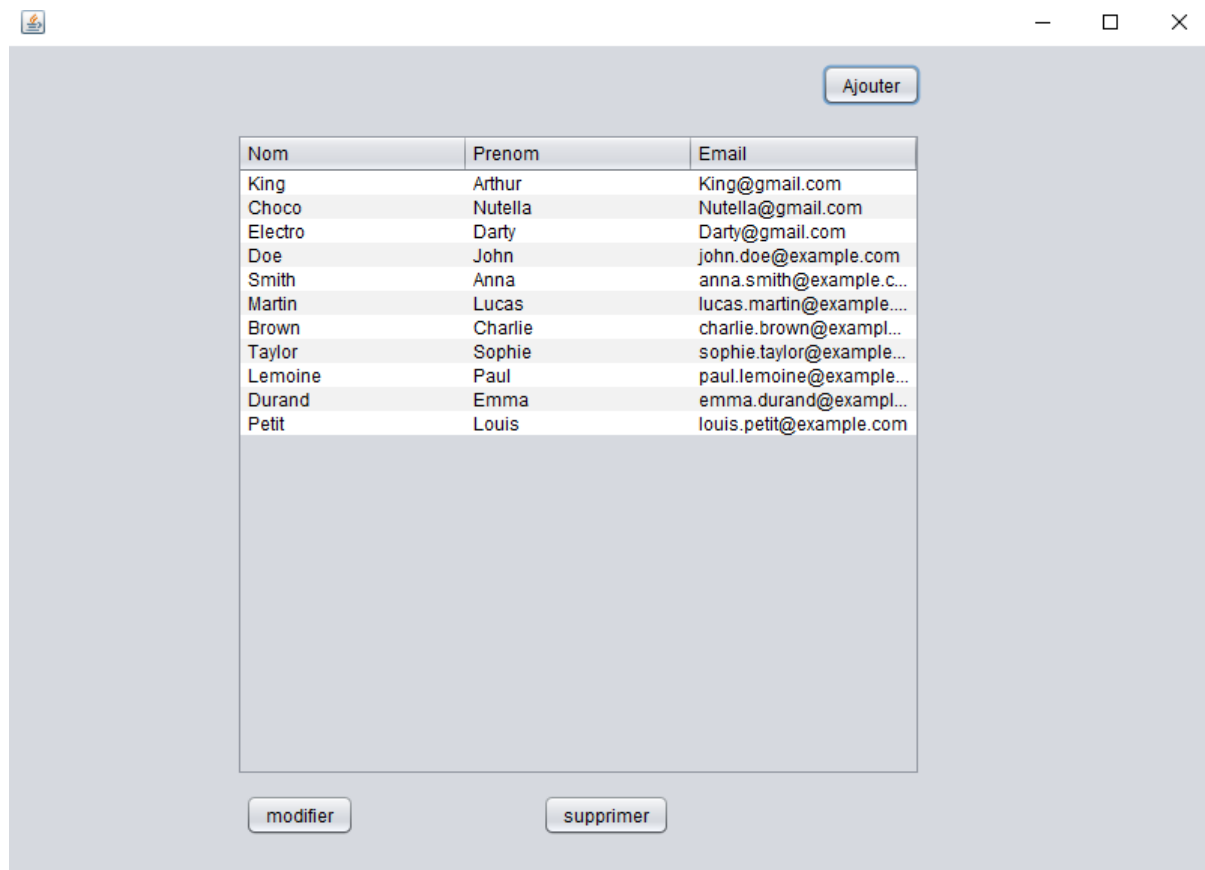
Ajouter

- Ou on peut choisir de l'ouvrir en invite de commandes en tapant « java -jar AP4.jar

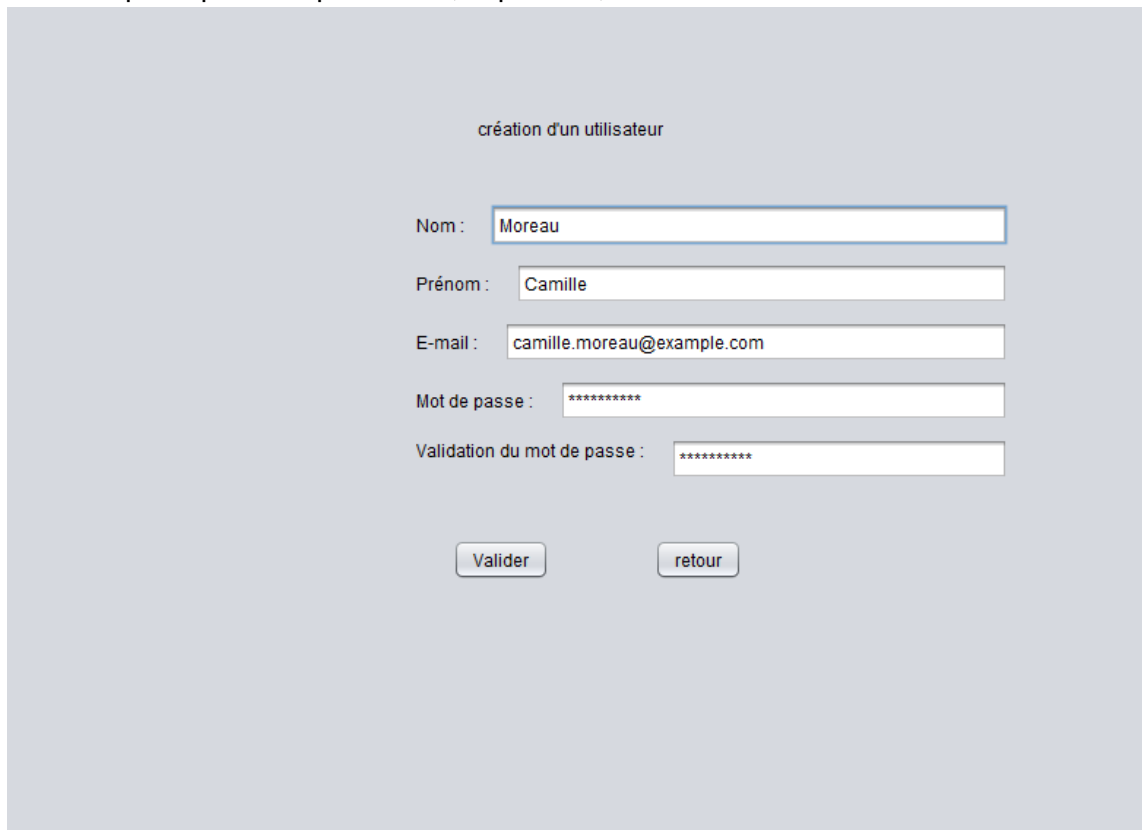


## Utilisation :

- Dès le lancement de l'application, la liste des utilisateurs est directement accessible.



- L'application permet ensuite d'ajouter, de modifier ou de supprimer un utilisateur. En cliquant sur "Ajouter", une fenêtre (AjoutDialog) s'ouvre, permettant de renseigner les champs requis tels que le nom, le prénom, et autres informations.



The screenshot shows a dialog box titled "création d'un utilisateur" (user creation). It contains several input fields for user information:

- Nom : Moreau
- Prénom : Camille
- E-mail : camille.moreau@example.com
- Mot de passe : [masked with asterisks]
- Validation du mot de passe : [masked with asterisks]

At the bottom of the dialog, there are two buttons: "Valider" (Validate) and "retour" (Return).

- Il faut rentrer le même mot de passe deux fois pour le valider sinon un message d'erreur s'affiche.

création d'un utilisateur

Nom :


Prénom :

E-mail :

Mot de passe :

Validation du mot de passe :

Erreur



Les mots de passe ne correspondent pas.

- Une fois l'ajout validé, le nouvel utilisateur apparaît dans la liste existante.

Nom	Prenom	Email
King	Arthur	King@gmail.com
Choco	Nutella	Nutella@gmail.com
Electro	Darty	Darty@gmail.com
Doe	John	john.doe@example.com
Smith	Anna	anna.smith@example.c...
Martin	Lucas	lucas.martin@example....
Brown	Charlie	charlie.brown@exampl...
Taylor	Sophie	sophie.taylor@example...
Lemoine	Paul	paul.lemoine@example...
Durand	Emma	emma.durand@exampl...
Petit	Louis	louis.petit@example.com
Moreau	Camille	camille.moreau@exam...

- Pour modifier un utilisateur, il suffit de le sélectionner dans la liste et de cliquer sur modifier.



— □ ×

Ajouter

Nom	Prenom	Email
King	Arthur	King@gmail.com
Choco	Nutella	Nutella@gmail.com
Electro	Darty	Darty@gmail.com
Doe	John	john.doe@example.com
Smith	Anna	anna.smith@example.com
Martin	Lucas	lucas.martin@example.com
Brown	Charlie	charlie.brown@example.com
Taylor	Sophie	sophie.taylor@example.com
Lemoine	Paul	paul.lemoine@example.com
Durand	Emma	emma.durand@example.com
Petit	Louis	louis.petit@example.com
Moreauto	Camilles	camilles.moreauto@gmail.com

modifier

supprimer

- La même fenêtre que celle utilisée pour l'ajout s'ouvre, mais cette fois-ci les champs de texte sont préremplis avec les informations de l'utilisateur sélectionné.

Modification d'un utilisateur

Nom :

Moreau

Prénom :

Camille

E-mail :

camille.moreau@example.com

Mot de passe :

\*\*\*\*\*

Validation du mot de passe :

\*\*\*\*\*

Valider

retour

- Il ne reste plus qu'à modifier les informations souhaitées et cliquer sur « Modifier » ; la mise à jour de l'utilisateur est alors immédiatement visible dans la liste.

Modification d'un utilisateur

Nom :

Prénom :

E-mail :

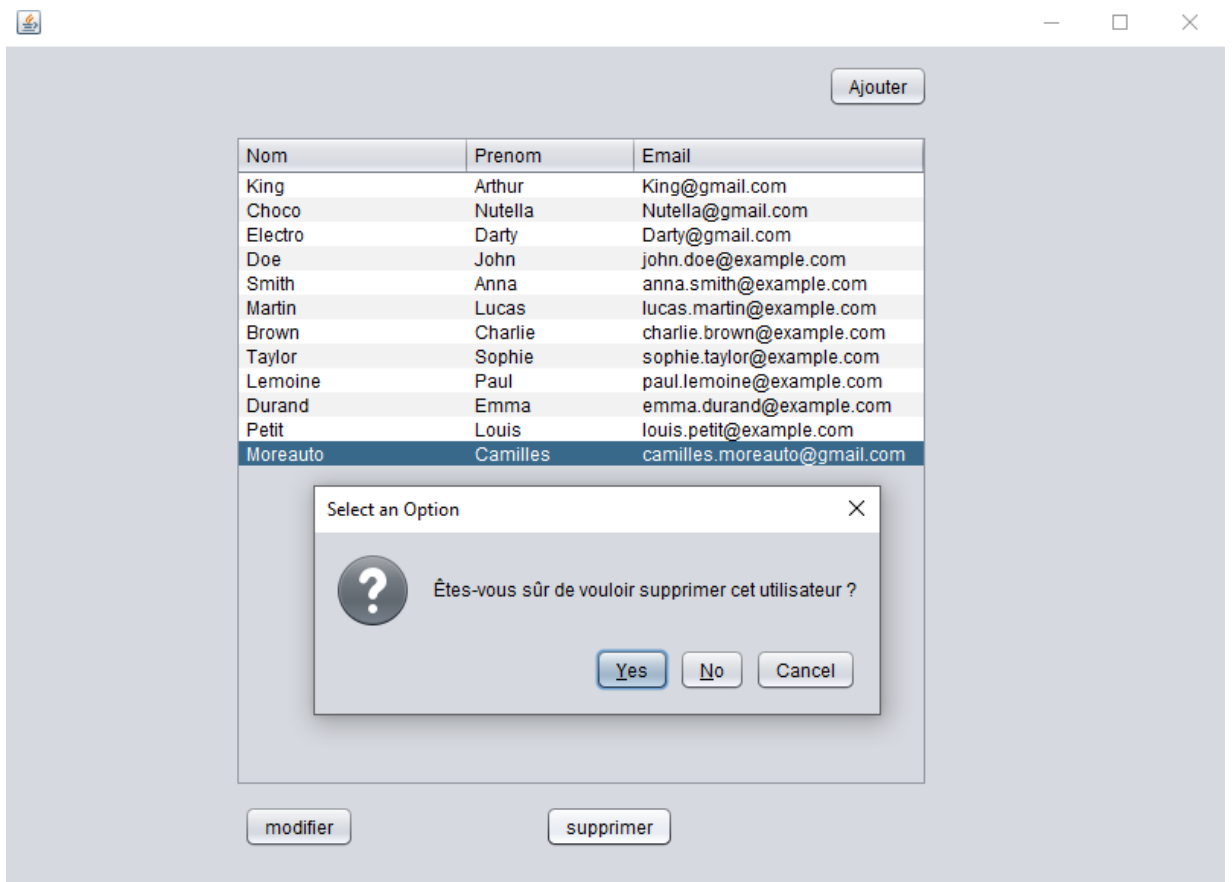
Mot de passe :

Validation du mot de passe :

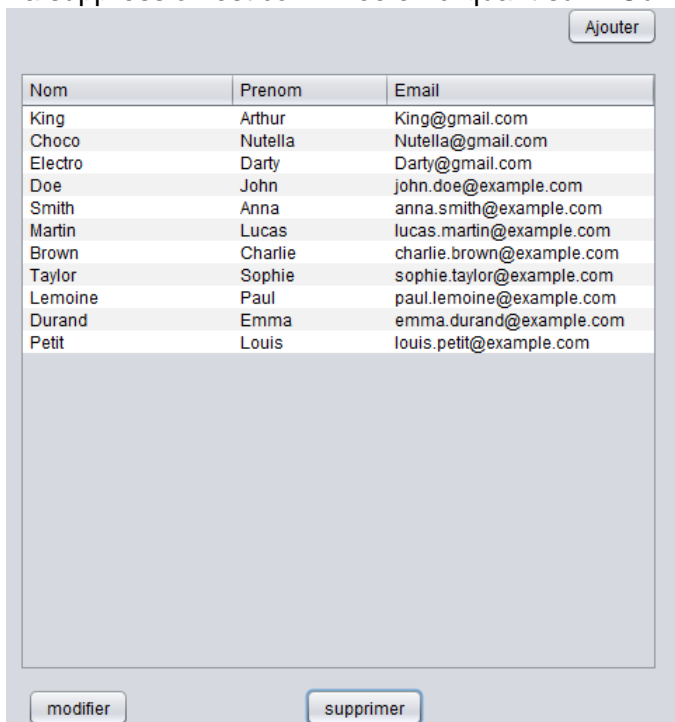
- Enfin, pour supprimer un utilisateur, il suffit, comme pour la modification, de sélectionner l'utilisateur souhaité dans la liste.

Nom	Prenom	Email
King	Arthur	King@gmail.com
Choco	Nutella	Nutella@gmail.com
Electro	Darty	Darty@gmail.com
Doe	John	john.doe@example.com
Smith	Anna	anna.smith@example.com
Martin	Lucas	lucas.martin@example.com
Brown	Charlie	charlie.brown@example.com
Taylor	Sophie	sophie.taylor@example.com
Lemoine	Paul	paul.lemoine@example.com
Durand	Emma	emma.durand@example.com
Petit	Louis	louis.petit@example.com

- En cliquant sur le bouton « Supprimer », un message de confirmation apparaît pour valider l'action de suppression.



- Si l'on clique sur « Non » ou « Cancel », la suppression est annulée
- La suppression est confirmée en cliquant sur « Oui »



## 7. Planning et gestion de projet

### Répartition des tâches :

La première étape a été la mise en place de GitHub, réalisée par Lewis, Eliaz et Manu.

La deuxième étape a consisté en la création du MCD et du script SQL, assurée par Lewis et Manu.

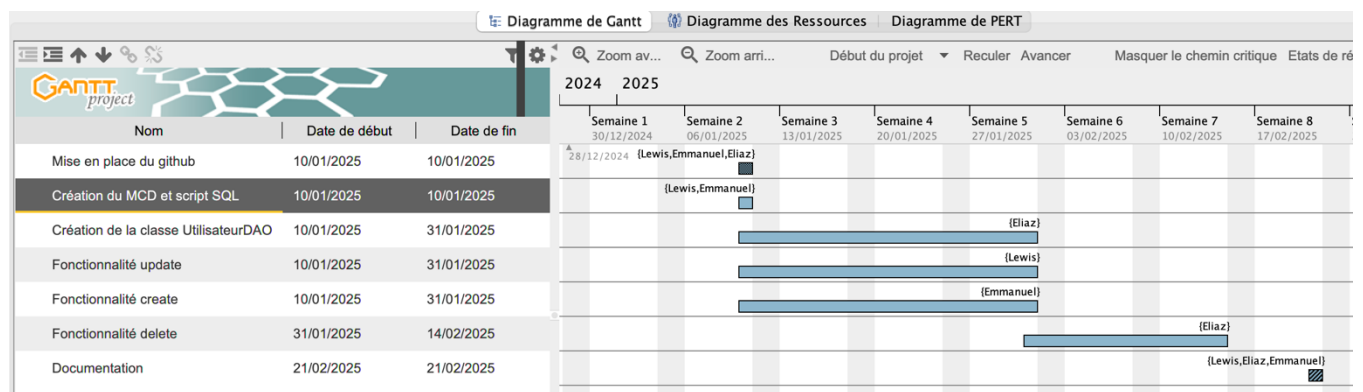
La troisième étape a été dédiée au développement de la classe UtilisateurDAO par Eliaz. Ensuite, Lewis a pris en charge la réalisation du formulaire de modification ainsi que de la méthode update.

En parallèle, Manu s'est occupé de la création du formulaire d'ajout et de la méthode create.

Eliaz a poursuivi avec le développement de la méthode delete, permettant la suppression des utilisateurs.

Enfin, la documentation servant de support pour l'oral a été rédigée en collaboration par Eliaz, Lewis et Manu.

### Diagramme de Gantt :



## 8. Problèmes rencontrés et solutions

Nous avons rencontré un problème avec GitHub sur NetBeans, où nos versions du projet se sont mélangées. Nous ne pouvions plus effectuer de push sans risquer d'écraser le travail de l'autre. Ce problème est survenu car nous ne maîtrisons pas parfaitement Git sur NetBeans, étant donné que c'était nos débuts avec cet outil. La solution résidait dans notre mauvaise gestion de la procédure : il nous fallait d'abord faire un pull avant de pouvoir push. Ce problème a tout de même constitué un obstacle majeur et nous a fait perdre un certain temps.

## 9. Bilan et conclusion

La réalisation de ce projet client lourd a été une expérience **très enrichissante** pour l'ensemble de notre équipe.

Elle nous a permis de renforcer notre **cohésion de groupe** et de développer des compétences solides en programmation orientée objet, en modélisation MVC, ainsi qu'en conception d'interfaces avec Swing sous NetBeans.

Chacun de nous a pu **progresser** techniquement et humainement malgré les quelques difficultés rencontrées, ce projet a été une **expérience formatrice, motivante et professionnelle**, qui nous prépare avec confiance aux défis du monde du travail.