*Bordji Nada – Lallali Amine*

*Rapport*

Gestion de projet des étudiants

Table des matières

[Conception du projet 2](#_Toc152549439)

[Installation des outils informatiques 2](#_Toc152549440)

[Utilisation d’une base de données ou non 3](#_Toc152549441)

[Diagramme de la base de données 3](#_Toc152549442)

[Schéma de l’interface graphique 3](#_Toc152549443)

[Réalisation du projet 6](#_Toc152549444)

[Hiérarchisation des classes (Data, Main, InterfaceGraphique..) 6](#_Toc152549445)

[Création des classes 6](#_Toc152549446)

[Création de l’interface graphique 7](#_Toc152549447)

[Problèmes rencontrés 7](#_Toc152549448)

[Installation de MySql sur les machines 7](#_Toc152549449)

[Connexion à la base de données 8](#_Toc152549450)

[Conception de l’interface graphique 8](#_Toc152549451)

[Conclusion 8](#_Toc152549452)

[ANNEXE 9](#_Toc152549453)

Rapport de Projet

Bordji Nada, Lallali Amine

Ce rapport rend compte de la démarche suivie dans la création d’une application destinée à la gestion de notes d’étudiants. Cette application, dont le fonctionnement est principalement dû à une interface graphique, doit pouvoir collecter les différentes données liées aux étudiants, aux formations, aux groupes de travail, aux projets à noter ou encore aux notes de chacun des étudiants.

# Conception du projet

## Installation des outils informatiques

Ayant décidé d’utiliser une base de données SQL, nous avons du, avant de commencer à coder sur le fond du projet, installer le logiciel MySQL. Nous avons travaillé sur ce projet sur deux systèmes d’exploitation différents (Mac OS et Windows) ce qui a rendu l’installation et l’utilisation de ce logiciel quelque peu compliqué.

Après avoir installer le logiciel nous permettant de créer notre base de données, nous devions trouver un moyen pour connecter notre code a celle-ci. Pour ce faire, nous avons installer un connecteur MySQL. Nous avons eu énormément de difficultés pour passer cette étape. Étant cruciale pour pouvoir démarrer le projet, nous avons consacré beaucoup de temps dans la recherche d’un moyen pour débloquer son passage. Dès lors que la connexion a été assurée avec le fichier java, nous avons commencé à développer notre application.

Nous avons décidé d’utiliser le logiciel Visual Studio Code par facilité puisque nous l’avions déjà utilisé auparavant. Ce choix est également justifié par la possibilité d’exécuter à la fois le code java et SQL sur la même interface grâce aux différentes extensions proposées par le logiciel. Nous pouvions donc exécuter les requêtes avant de les intégrer à notre code pour vérifier que les résultats qui en ressorte sont bien cohérents.

Afin de mettre en commun notre travail de manière régulière et sans avoir à forcément se voir pour se mettre à jour concernant l’évolution de l’application, nous avons utilisé la plateforme GitHub sur laquelle on déposait toutes nos mises à jour. La plateforme nous permettait la synchronisation continue sur nos avancées respectives de manière assez simple et intuitive, notamment grâce à leur application Bureau.

## Utilisation d’une base de données ou non

Comme mentionné plus tôt, nous avions eu beaucoup de difficultés à connecter notre base de données à notre code. Ce blocage nous a fait douter de l’utilisation même de cette base de données. On avait pensé, dans le cas échéant, à utiliser des classes instanciées qui correspondraient à chacune des tables que l’on avait initialement prévu de mettre dans notre base de données.

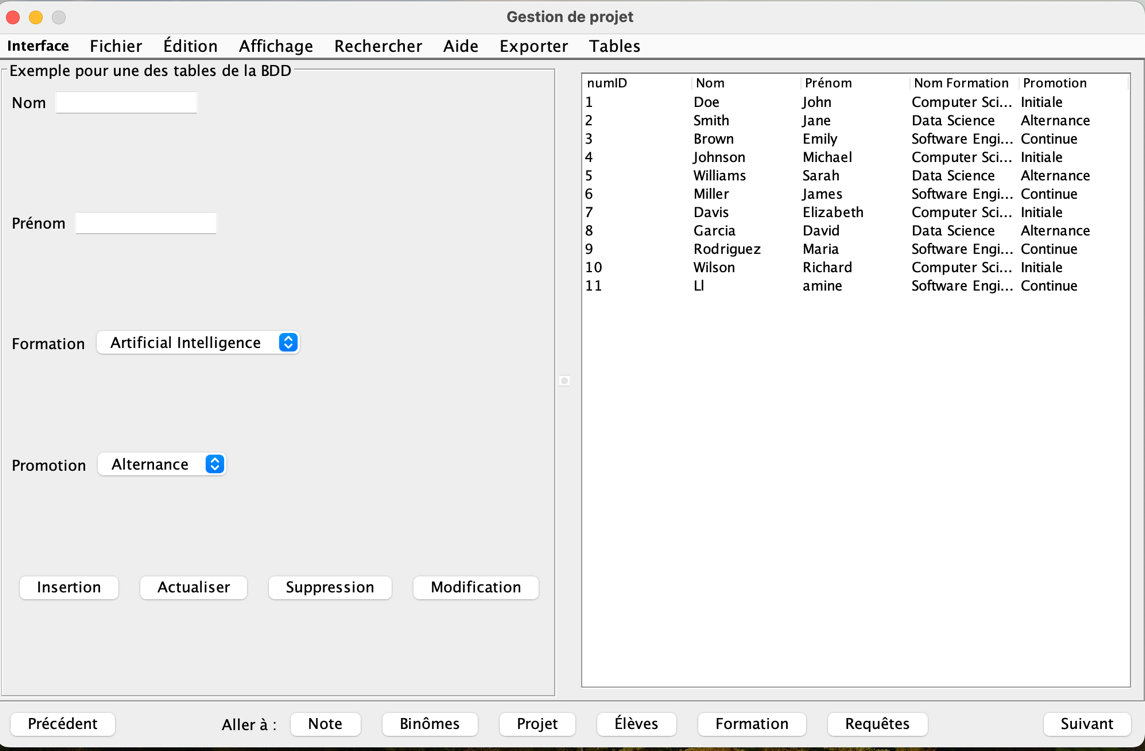
Après réflexion, nous avons fini par trouver que cette solution risquait d’être très couteuse en termes d’espace et rendrait notre code beaucoup moins lisible que ce que nous imaginions au départ. C’est pour cela que nous nous étions particulièrement acharnés pour trouver une manière d’utiliser la base de données et cela a fini par porter ses fruits.

La base de données nous a permis de centraliser l’ensemble des données que nous avions à disposition et de mettre à jour ces données qui sont souvent entreliées sans avoir à modifier chacune de leurs occurrences dans les différentes tables. Globalement, la base de données a permis de faciliter la centralisation et la gestion globale des données qui etaient ou seront entrés par l’utilisateur.

## Une image contenant texte, capture d’écran Description générée automatiquementDiagramme de la base de données

## Schéma de l’interface graphique

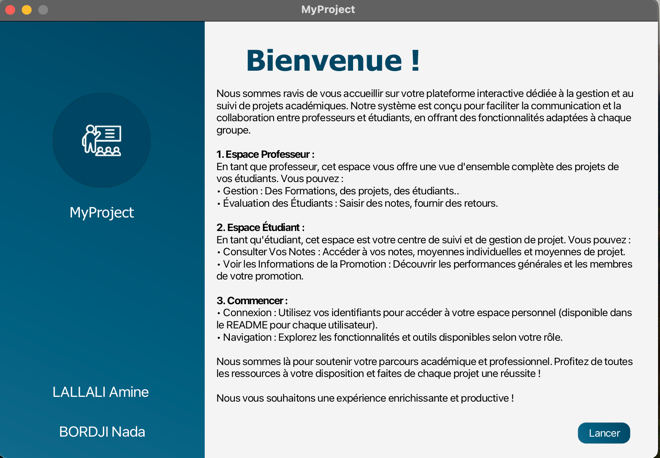
Notre première approche dans la création de l’interface graphique s’était faite en utilisant la bibliothèque Swing que nous avions déjà utilisé et que nous maitrisions en partie. Néanmoins, en discutant avec les autres groupes, nous avons vu qu’elle ne permettait pas d’avoir beaucoup de choix sur le design de l’interface. Nous avons donc décidé d’utiliser la bibliothèque Java FX et le logiciel SceneBuilder. Ces deux éléments nous ont laissé beaucoup plus de choix dans la conception de notre application.

Dans un premier lieu, nous avons tout de même commencer le développement de notre application de Gestion de *Gestion de Projets* avec la bibliothèque Swing. Cela nous a pris une bonne partie de notre temps (environ 1 mois). En effet, notre première application était très « primaire ». Aucunes couleurs apparentes, aucunes formes, aucunes animations dynamiques en actionnant un bouton. Nous avons tout de même implémenté toute notre application en ayant « fini » le projet. Voici un exemple de notre version 1.0 :

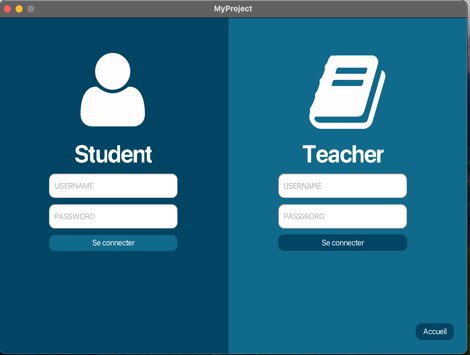
Comme vous pouvez le voir, elle ne répondait pas entièrement à nos attentes en termes d'esthétique et d'interaction utilisateur. L'interface était fonctionnelle, mais manquait de modernités. Les boutons étaient basiques et le tableau de données, bien que fonctionnel, avait un style très standard sans personnalisation. Les menus étaient purement fonctionnels, avec des titres génériques comme "Fichier", "Édition", "Affichage", qui ne reflétaient pas l'aspect intuitif que nous visions. Malgré ces limites, cette première tentative était essentielle pour définir les fonctionnalités de base et la logique métier de notre application. Elle nous a permis de structurer notre code et de préparer le terrain pour une transition en douceur vers JavaFX, où nous pourrions ensuite peaufiner l'interface et l'expérience utilisateur.

En effet, en adoptant Java FX, le travail de base sur la structure des classes et la connexion à la base de données avait déjà été fait. On avait le contenu, mais il nous manquait l'aspect visuel. Cependant, avec SceneBuilder, on a réussi à passer à la vitesse supérieure assez vite.

SceneBuilder, c'est un outil qui nous a facilité plusieurs étapes dans la conception de l'interface utilisateur. Il nous a permis de glisser et déposer des composants pour assembler nos écrans, sans qu'on ait besoin de taper tout le code à la main. Nous pouvions voir les changements en temps réel et ainsi modifier le design.

Voici une brève description de notre interface :

Pour commencer, après la compilation du projet et l’exécution du fichier main, une première fenêtre de présentation s’ouvre. Lorsque l’utilisateur a pris connaissance de toutes les informations, il peut passer à la page suivante qui lui propose de se connecter en tant que professeur ou en tant qu’étudiant. Il pourra alors entrer un identifiant et un mot de passe qui seront fournis dans le READMIS.



Dans le cas où il se connecte en tant que professeur, la fenêtre permettant d’entrer les notes de chaque élève s’ouvrira en premier. Sur cette page, il aura la possibilité d’ajouter, supprimer, ou modifier les notes de soutenance et de rapport des étudiants par projet effectué. Un tableau s’affiche dans le bas de la fenêtre et affiche l’ensemble des notes qui ont été entrées.

Pour passer à une autre catégorie, l’utilisateur doit appuyer sur le menu se situant en haut à gauche de la page et qui ouvriront une liste déroulante contenante toutes les pages nécessaires dans la gestion de projet.

Les autres catégories mentionnées sont celles des formations, des étudiants, des projets et des binômes. Leurs pages respectives sont construites de la même manière avec la possibilité d’ajouter grâce aux zones de textes et aux listes déroulantes correspondant à chaque caractéristique, de supprimer ou de modifier un élément de chacune de ces tables, la possibilité de visionner les tables dans le bas de la page.

Une dernière page peut être visionnée à partir du menu déroulant se trouvant en haut à gauche de la fenêtre : celle du tableau de bord. Cette page permet au professeur d’avoir une vue d’ensemble sur les projets des élèves. On y retrouvera notamment un compteur permettant de savoir combien de projets sont en cours et peuvent donc encore être rendus à temps, le nombre de binômes ayant rendu un projet en retard ou encore les différents sujets de projet proposés par chacune des formations. Les informations trouvées sur ces pages permettent en premier lieu au professeur d’avoir certaines statistiques sur ses étudiants et lui éviteront de perdre du temps à tenir un carnet précis qui contiendra ces informations.

# Réalisation du projet

## Hiérarchisation des classes (Data, Main, InterfaceGraphique..)

Concernant la hiérarchisation des classes nous avons fait face à plusieurs problèmes. En effet, notre code ne possède pas vraiment de hiérarchie ou encore de polymorphisme. Notre stratégie est d’éviter le traitement de données (plus ou moins). Elle consiste à récupérer les données soumis par l’utilisateur puis à les insérer sans passer par l’intermédiaire de constructeur, toutes nos classes n’étaient donc pas instanciable. Malheureusement, cette stratégie n’a pas pu supporter la migration vers Java FX. L’instanciation des différentes classes étaient nécessaire voir indispensable. Java Fx demandait la manipulation d’objet, ce qui nous amené à revoir notre code, de A à Z (Binôme, Student, formation…) Ainsi, nous avons dû mettre en place des classes indépendantes sans hiérarchisation particulières par faute de temps ce qui entraine le développement avec un code « lourd ». C’était l’une des principales difficultés de ce projet malgré la simplicité du sujet, la gestion.

## Création des classes

Nous avons décidé de structurer notre projet grâce à 4 packages distincts :

* Le package Data qui contient la méthode permettant de se connecter à la base de données et la méthode d’initialisation de la base de données ainsi que la base de données elle-même
* Le package InterfaceGraphique contenant toutes les classes Controller liées aux fichiers FXML de l’interface Graphique. Ainsi que les fichiers CSS contenant le style des composantes de l’App.
* Le package Model qui contient toutes les classes associées à chacune des tables de la base de données
* Le package Main qui contient le fichier Main du projet qui permet l’exécution du projet

Les fichiers java dans le package MODEL contiennent chacun les méthodes propres à la classe dans lesquelles on les retrouve. Par exemple, dans la classe Formation, on trouvera une méthode permettant l’ajout d’une formation à la base de données, une autre permettant la suppression d’une formation et une autre permettant de mettre à jour la table de Formation. Ces 3 méthodes sont communes (dans la fonction) à l’ensemble des classes représentant une table dans la base de données. Le package MODEL regroupe ainsi les classes permettant l’interaction avec la BDD.

## Création de l’interface graphique

Comme indiqué précédemment, nous avons opté pour JavaFX comme Framework pour développer l'interface graphique de notre application. Cette bibliothèque offre une grande flexibilité et une variété de fonctionnalités qui sont essentielles pour créer des interfaces dynamiques et interactives. Pour faciliter et accélérer le processus du développement de l’application, nous avons utilisé Scene Builder.

Scene Builder est un outil visuel qui permet de construire des interfaces graphiques en glissant et déposant des composants UI, sans nécessiter une écriture de code manuelle pour la mise en page. Cela rend le processus de conception beaucoup plus intuitif et accessible, surtout pour ceux qui sont moins expérimentés en codage comme nous.

Pour maîtriser cet outil, nous nous sommes appuyés sur plusieurs tutoriels disponibles sur YouTube (cf. Marcoman). Ses tutoriels couvrent une gamme étendue d’application mais avaient plus ou moins la même forme. L'utilisation de Scene Builder en combinaison avec les tutoriels YouTube nous a présenté plusieurs avantages :

* **Rapidité de Développement**
* **Apprentissage Facilité**
* **Flexibilité et Personnalisation**
* **Intégration Facile avec le Code**

# Problèmes rencontrés

## Installation de MySql sur les machines

Nous l’avons mentionné au début de ce rapport, la connexion à la base de données s’est montrée beaucoup plus difficile que ce que nous imaginions. Le premier souci que nous avions rencontré était dans l’installation de l’extension MySQL sur VS Code et dans la création d’une connexion. Une erreur revenait sans cesse, « ER\_ACCESS\_DENIED\_ERROR : Access denied for user 'root'@'localhost' ». De plus, nous travaillons sur des systèmes d’exploitation différents, donc les méthodes étaient assez différentes.

Plusieurs tutoriels et passages sur de nombreux forums plus tard, nous avions enfin trouvé la solution adaptée à nos systèmes d’exploitation respectif en reconfigurant nos logiciels MySQL et en passant directement sur nos terminaux pour pouvoir créer des comptes avec le bon « host » et le bon « username ».

## Connexion à la base de données

L’étape de l’installation MySQL étant passée, nous devions maintenant nous attaquer à la connexion à la base de données. Plusieurs forums sur internet dirigeaient vers l’installation du connector mysql et de l’ajout de ce fichier jar aux bibliothèques référencées par le projet java. Néanmoins, cette action n’a pas suffi à nous accorder la connexion à notre base de données.

Nous ne trouvions pas d’autres solutions à ce problème sur internet, tous les sites menaient à la même solution qui est celle que nous avions déjà appliqué. Après plusieurs jours de réflexion dessus et après avoir consulté les autres groupes ayant pour devoir ce projet, en vain, nous avons fini par faire des tests, des tentatives qui incluait l’utilisation du contenu du fichier jar que nous avions installé.

Nous savions qu’au final notre code devait trouver un moyen d’accéder à la classe référencée grâce au chemin relatif « com.mysql.cj.jdbc.Driver ». Le dossier com se trouvant dans le jar installé, nous avons essayé sans espoir de l’extraire et de le mettre dans le dossier de notre projet et nous avons été assez surpris de voir qu’après cela notre connexion se faisait enfin.

## Conception de l’interface graphique

Malgré une conception assez « simple » et intuitive nous avons tout de même rencontré plusieurs problèmes lors du design de l’application. En effet, le design n’était pas une tâche si facile que ça, car cela demandait de la créativité, de la précision mais aussi de la réflexion. Tout d’abord, nous avons fait face à plusieurs problèmes. La prise en main du logiciel était une étape assez longue. Nous ne savions pas quelle composante utiliser (AnchorPane,StackPane…) . Nous avons dû s’inspirer et apprendre de visuels pris de sites afin d’avoir un rendu assez conforme.

De plus, lors de l’implémentation des actionneurs et des affichages, nous avons fait face à différents problèmes lors de la compilation tels que la gestion d’Exception, de ClassNotFound dû à des faux référencements dans le fichier FXML ou bien dans les fichiers Controller.

## Conclusion

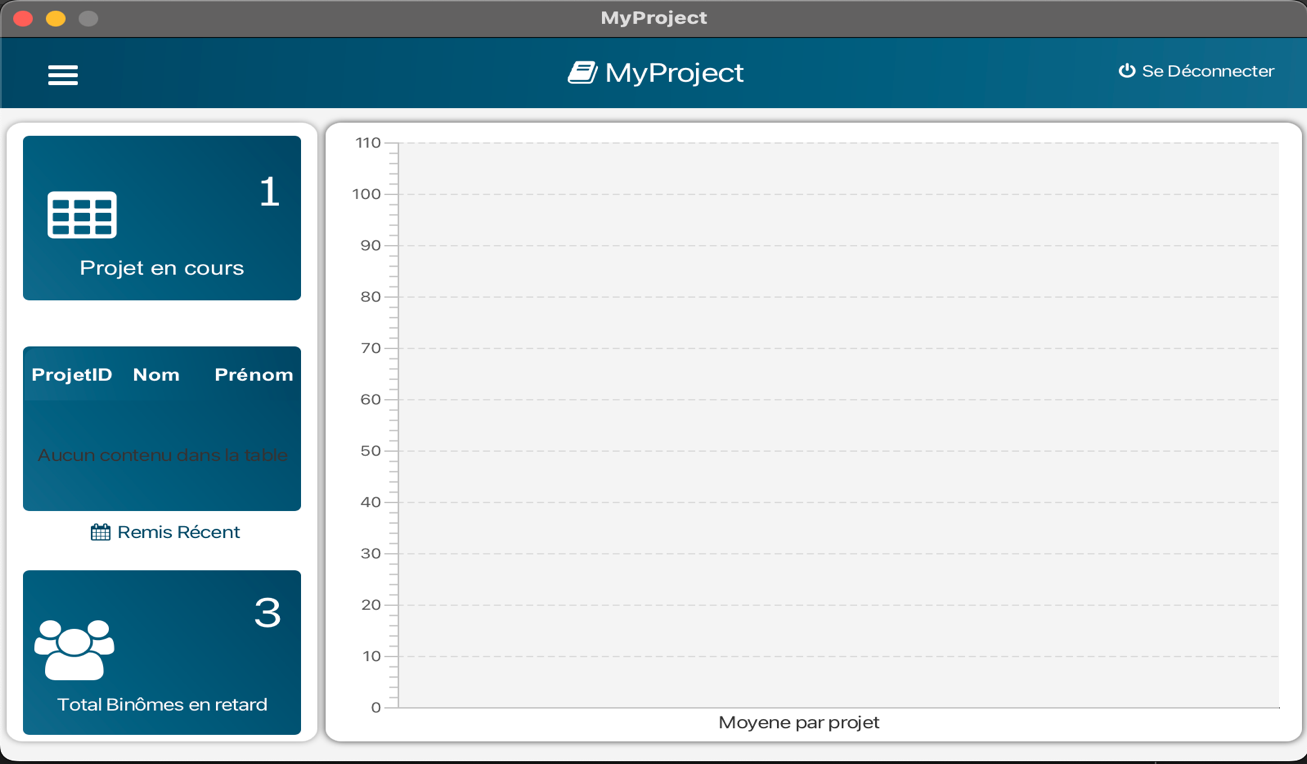
En conclusion, ce projet de développement d'une application de gestion de notes d'étudiants a été un parcours enrichissant et instructif. Malgré les défis techniques liés à l'utilisation de MySQL, JavaFX, Scene Builder, et la collaboration via GitHub, nous avons réussi à surmonter ces obstacles. L'expérience acquise dans la résolution de problèmes complexes, la conception d'interface utilisateur et la gestion de données est atout considérable pour notre insertion dans le monde du travail. Ce projet a non seulement renforcé nos compétences techniques, mais a également amélioré notre aptitude à collaborer et à s'adapter aux défis du développement logiciel.

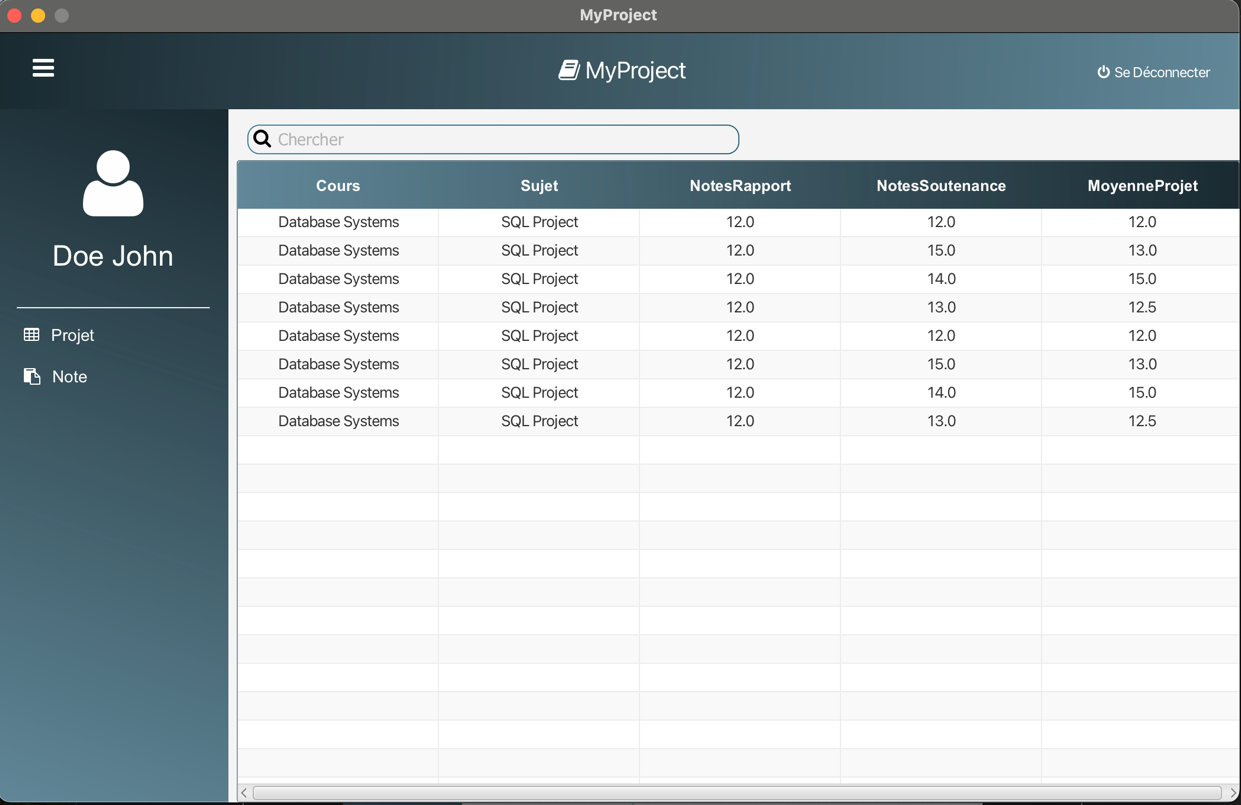
## ANNEXE

Chaine Youtube : [https://www.youtube.com/@marcomanchannel](mailto:https://www.youtube.com/@marcomanchannel)

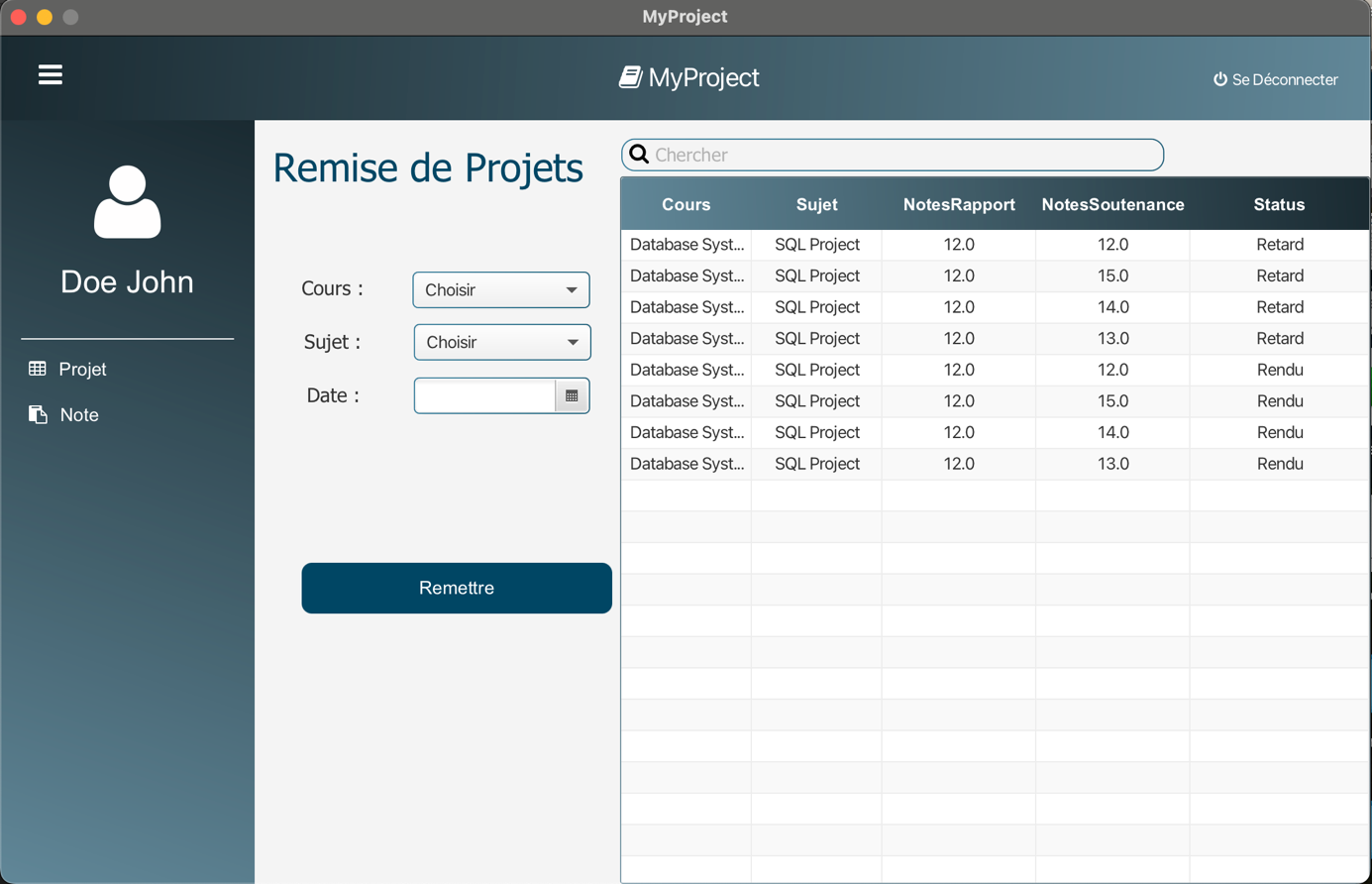
Chatgpt : [https://chat.openai.com](mailto:https://chat.openai.com)

FontAwesome : [https://fontawesome.com/v4/icons/](mailto:https://fontawesome.com/v4/icons/)



Exemple : Tableau de Bord

Exemple : Page Student page Note

Exemple : Page Student remise projet