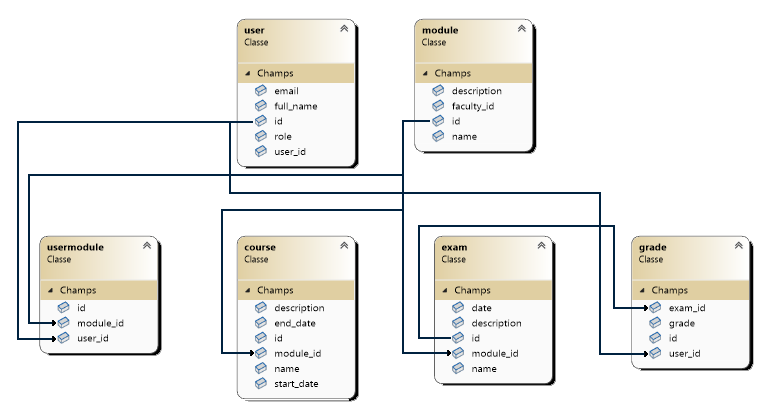
Final Assignment – Virtual Global College

Lucile Pelou Student n°74526

**Git repository:**

Pour cet assignement nous pouvions utiliser n’importe quel langage de programmation et n’importe quelle méthode pour arriver à faire l’exercice. Ayant déjà manipuler Firebase avec Node.js et javascript c’est vers cette méthode que je me suis tourné. J’ai donc tout d’abord créé un projet Firebase avec mes différentes collections :

**(Rajouter la fleche qui lie faculty id de module à id de user) + Grade dans module**

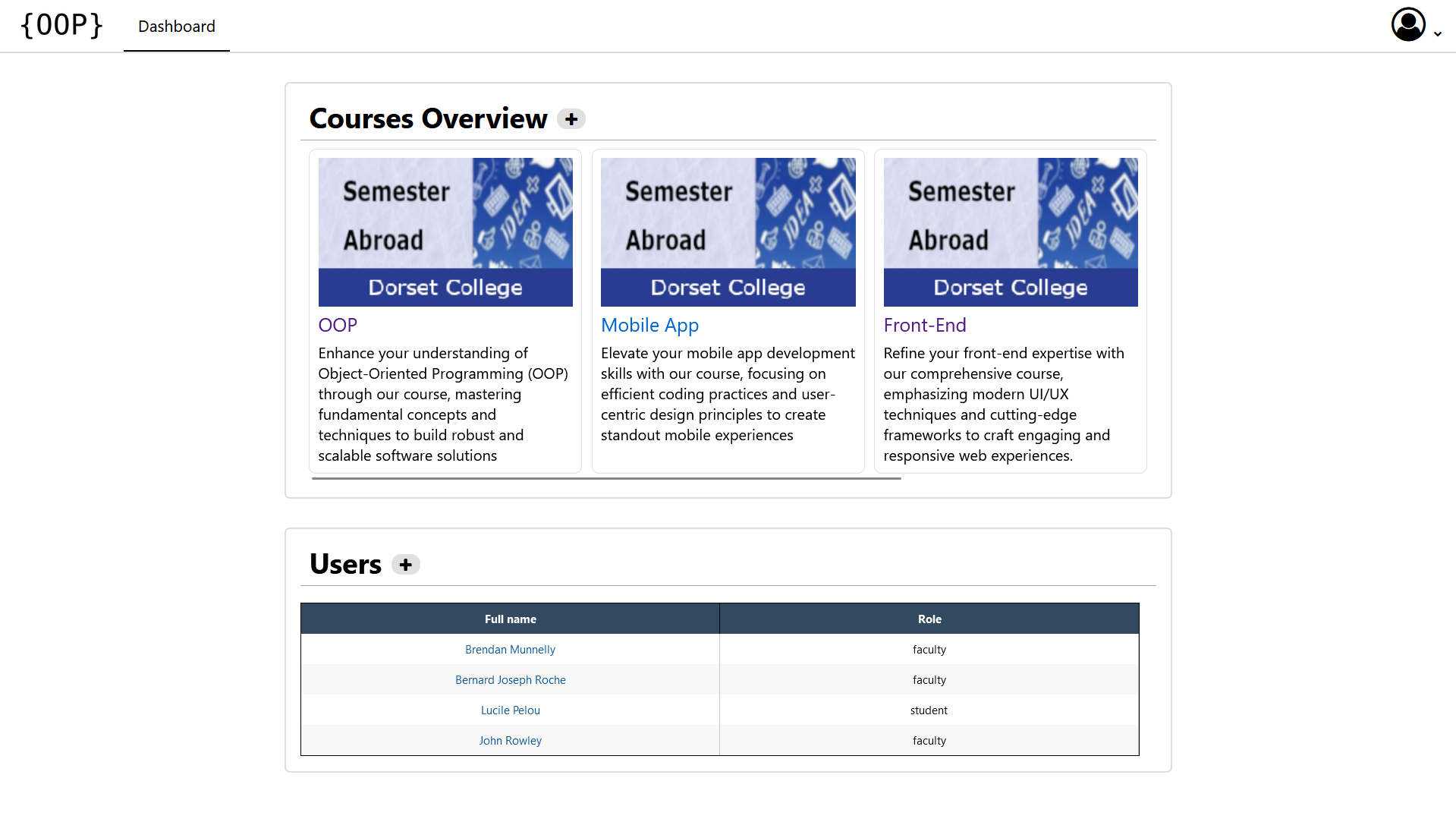
J’ai créé 6 collections comme le montre le diagramme si dessus. Pour gérer les utilisateurs et leurs authentifications j’ai utilisé l’authentification de Firebase qui fait ça très bien. Le champ id dans user\_id fait donc référence à l’id de l’utilisateur dans authentification et le mot de passe et l’email sont géré par l’authentification. J’ai ajouté le champ email dans la collection des users pour que cela soit plus simple.

Après avoir fait ça j’ai commencé mon projet en créant un git et ma première page : login. Cela m’a permis de pouvoir voir si l’authentification ainsi que la collection user fonctionnaient. J’ai également fait la méthode de recovery password dans le cas ou l’utilisateur l’a oublié, Firebase Authentification a déjà une méthode en place pour faire cela.

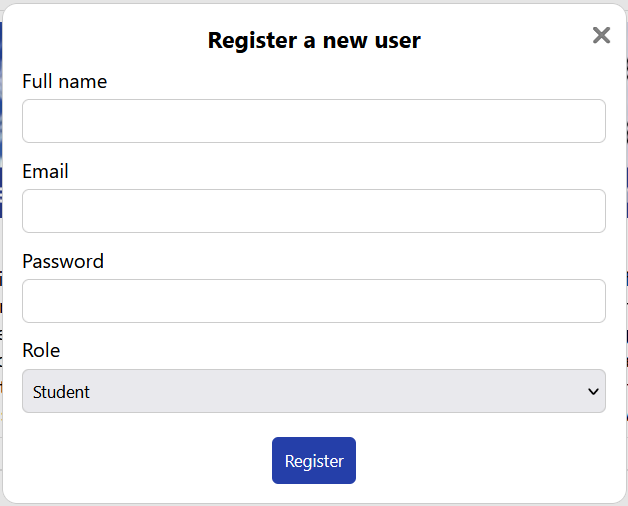
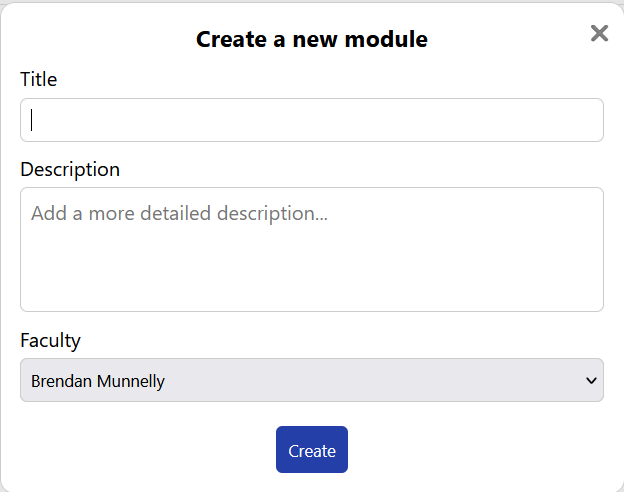
Au départ j’ai créé un fichier global.js ou j’ai mis toutes les collections ainsi que les noms des champs pour pouvoir les appeler et les changer plus facilement si besoin. Comme les collections sont appelé avec un string qui ne change pas dynamiquement avec la base de données, il était important que cela soit mis en place pour plus de facilité dans l’utilisation des différentes collections.

Après avoir fait ça je me suis lancé dans la construction des pages d’une personne ayant le rôle admin. Pour faire ça j’ai créé un dossier admin et une page admin\_dashboard ou j’ai commencé à mettre en place la logique de visions des différents modules et utilisateurs. Je me suis rendu compte par la suite que si je faisais une page par rôle cela allait vite devenir beaucoup et que je devrais avoir pleins de fichiers avec certaines fois les mêmes méthodes ou un fichier global trop volumineux et non lisible ce qui n’est pas très ingénieux. J’ai donc changé d’optique et ait décidé de faire une page dashboard pour tout le monde et que le contenu allait se changer dynamiquement en fonction du rôle de l’utilisateur. Cette approche permet d’avoir un espace de travail beaucoup mieux rangé. J’ai appliqué ça pour toutes les autres pages que j’ai fais par la suite. Pour pouvoir connecter la base donnée avec mon site j’ai utilisé toutes les méthodes tel que getDoc, updateDoc, … importé depuis Firebase.

1. Vue Administrateur
   1. Dashboard



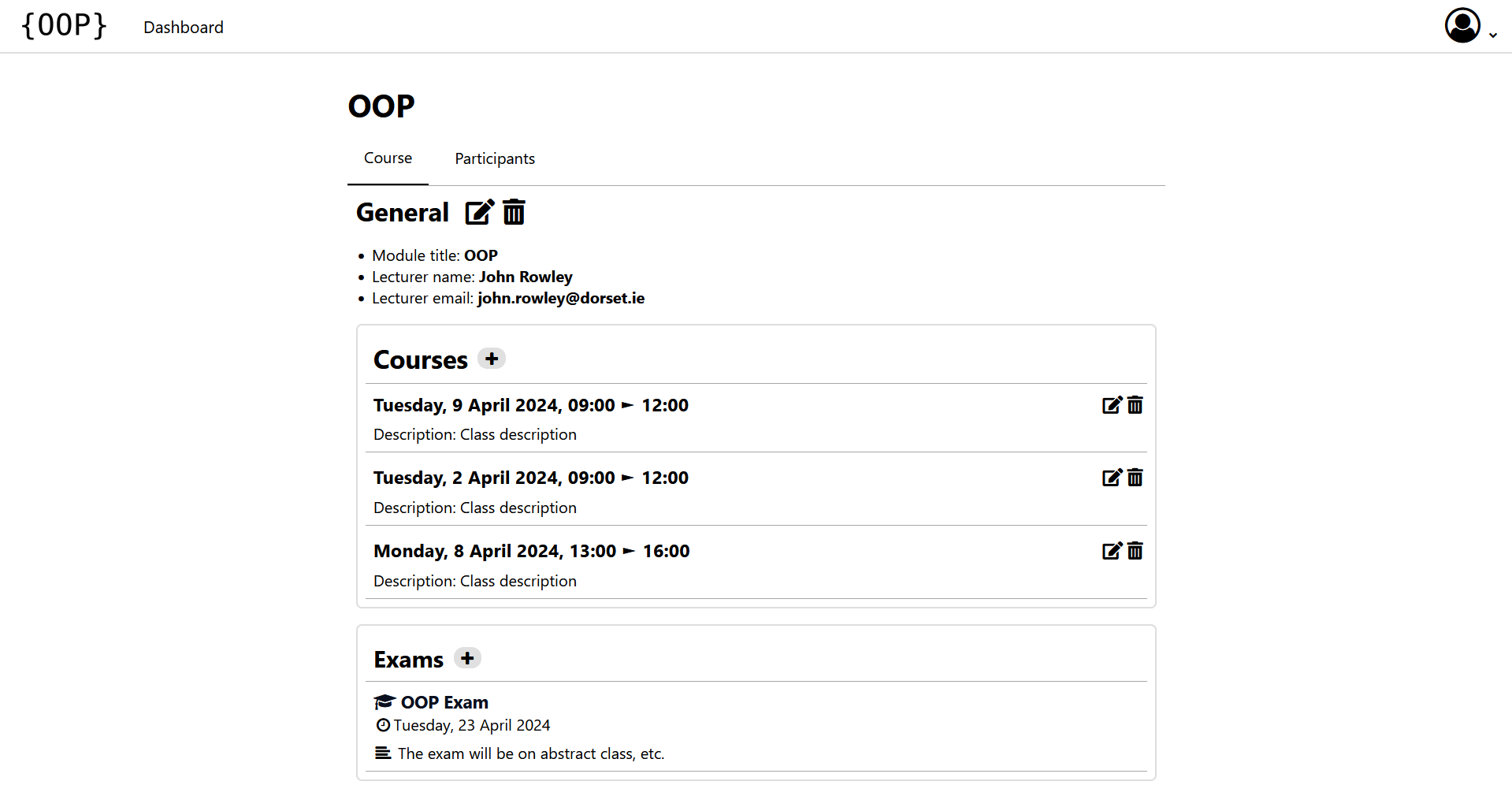
Sur cette page l’administrateur peut voir tous les modules ainsi que la liste des utilisateurs. Il peut également ajouter des modules et des utilisateurs en cliquant sur les boutons plus :



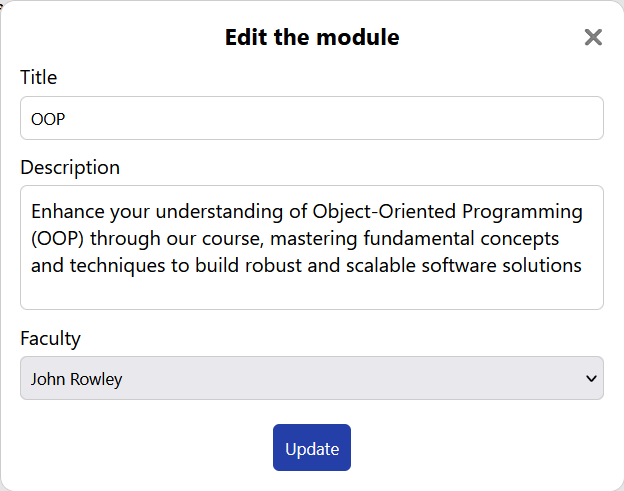
Ces boutons ouvrent des dialogues permettant à l’administrateur de rentrer les informations pour créer ces nouveaux objets.

Il peut également cliquer sur les différents pour accéder à leur page et voir leurs informations.

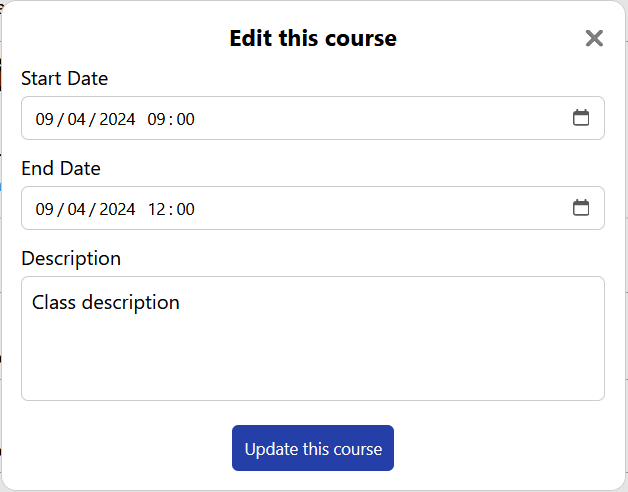
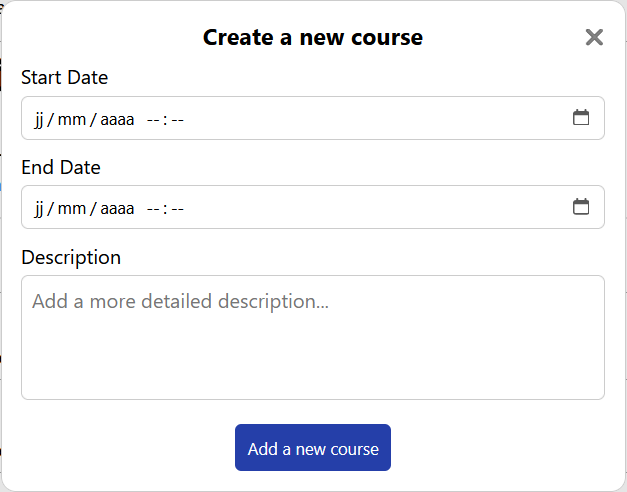
* 1. Module



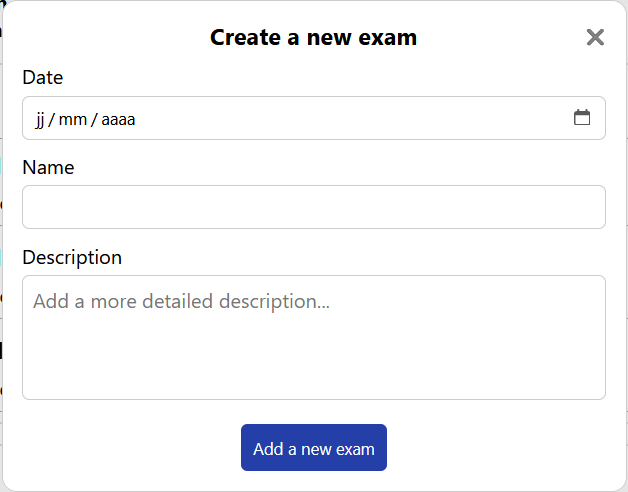
Sur cette page, l’administrateur peut voir les informations et objets rattaché au module. Il y a deux pages, course et participants. Dans la première il y a tout d’abord les informations du module tel que le titre, le nom du professeur ainsi que son mail. En dessous il y a les cours ainsi que les examens du module. Tout en haut, à côté de général il est possible de cliquer sur l’icône d’édition pour pouvoir éditer le module :



La corbeille lui permet de le supprimer (avec tous les objets qui y sont rattachés). Dans les cours il peut en ajouter un nouveau, éditer et supprimer ceux déjà présent.

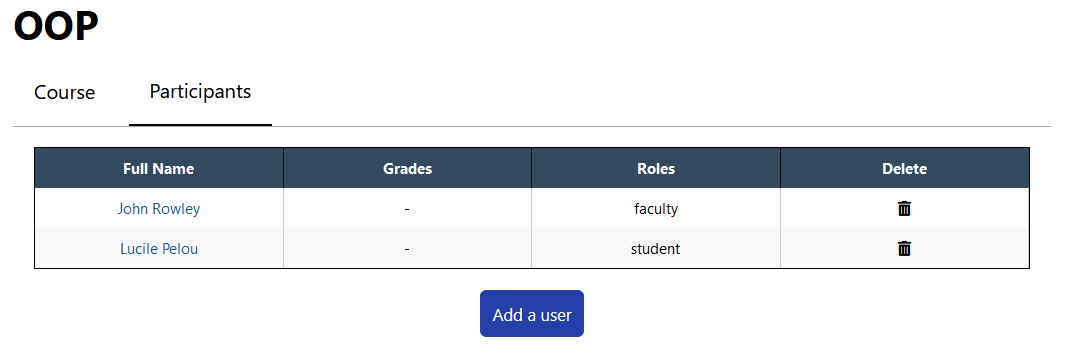


Pour les examens il peut en créer des nouveaux en cliquant sur le bouton plus :



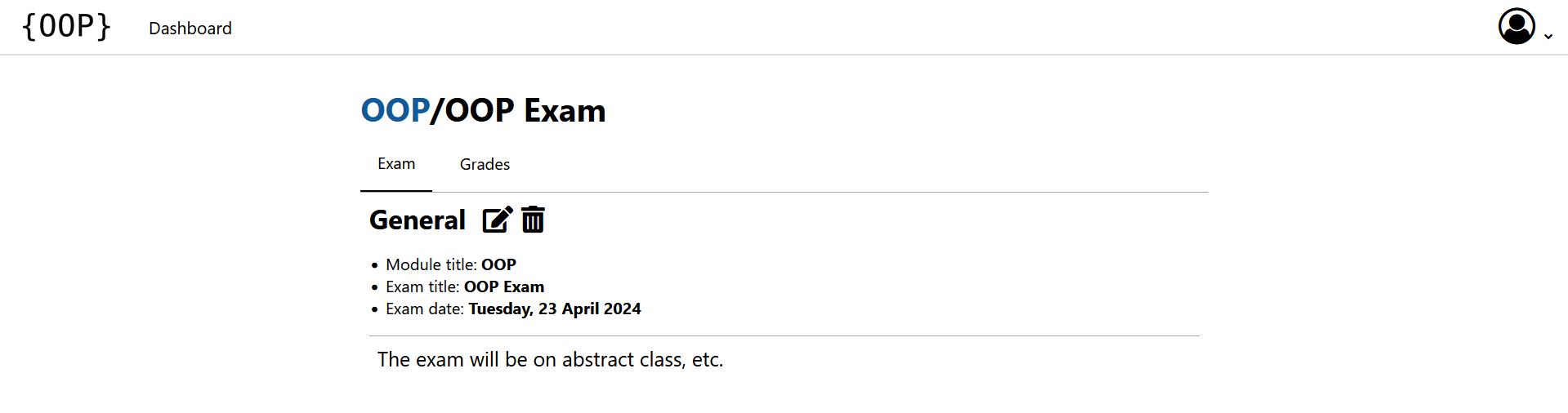
Et il peut également aller voir les informations de ceux déjà créer en cliquant sur leurs titres.

Le page participant lui permet de voir tous les utilisateurs rattachés à ce cours ainsi que leur note. Il peut en ajouter et en supprimer. Il ne peut pas éditer la note puisque celle-ci est la combination de tous les examens.

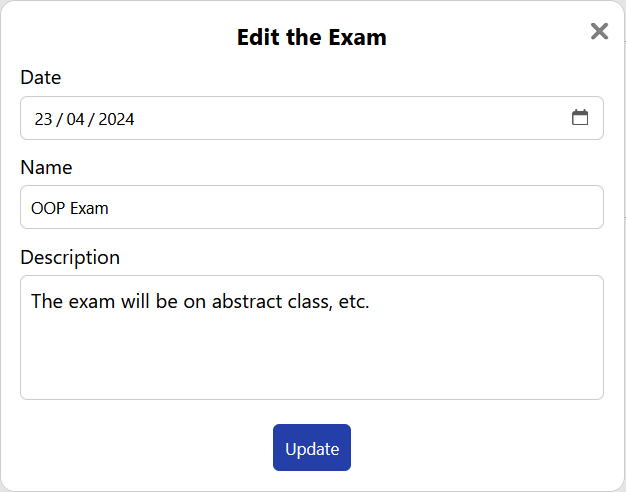


* 1. Exam

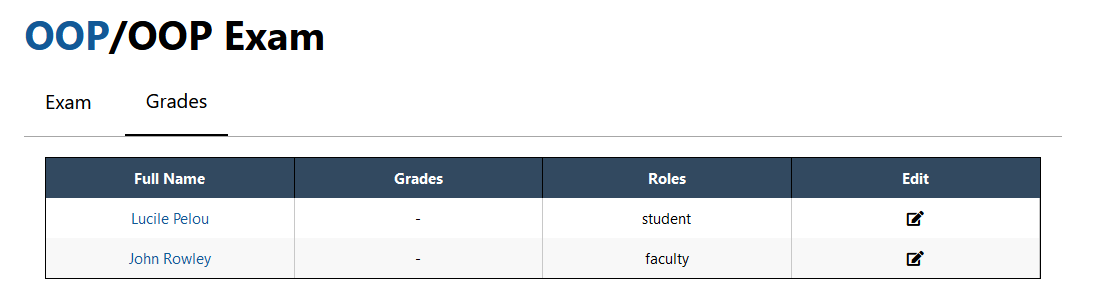
Comme dit précédemment l’admin peut cliquer sur un titre d’examen dans la page module pour voir ses informations.



Ici il peut voir les informations comme le nom du module, le titre de l’examen, sa date et sa description. Comme pour les modules il peut l’éditer et le supprimer.

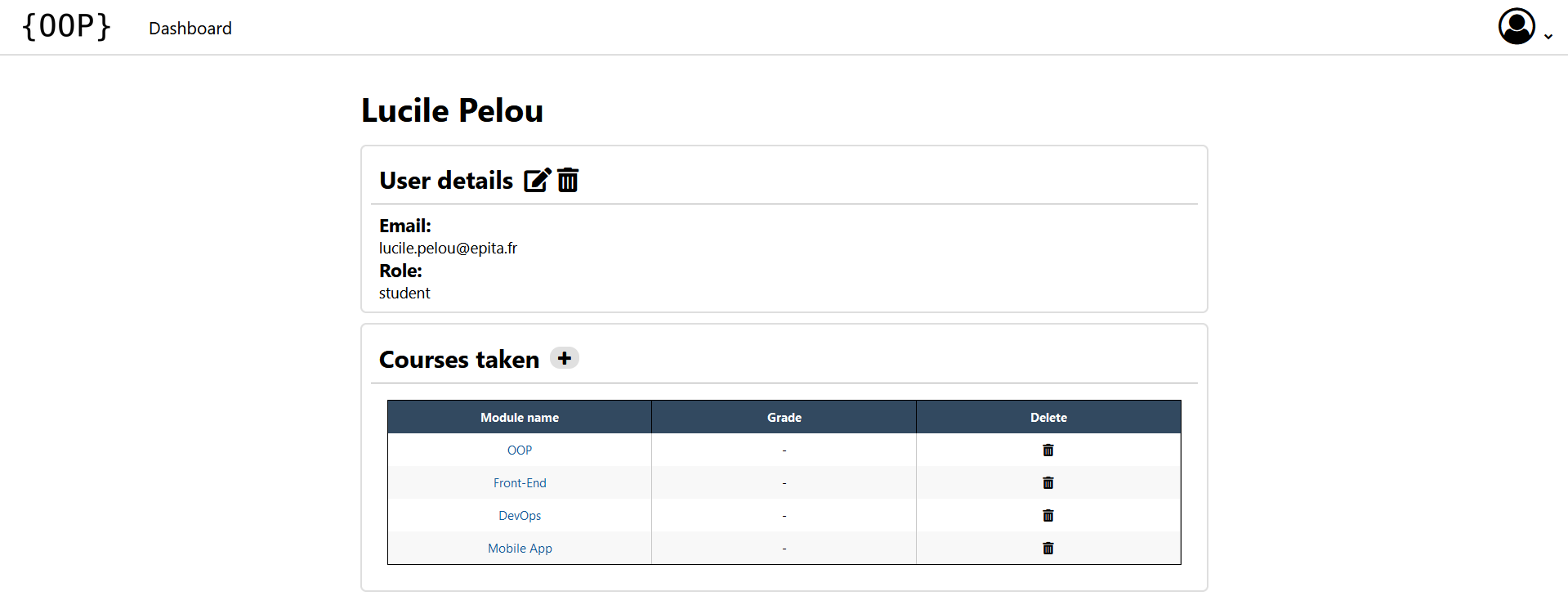


Il peut accéder à la page Grades qui va lui montrer les participants à l’examen et leurs notes.



* 1. Profile

L’admin n’a pas de profile étant donné qu’il n’a pas de cours donc ça n’a pas d’intérêt. Il peut par contre aller voir le profil des utilisateurs, par exemple en cliquant sur l’un des prénoms sur le tableau de l’image précédent il peut accéder à son profile.



D’ici il peut voir les informations concernant l’utilisateur et modifier celle-ci. Il peut également lui ajouter ou lui supprimer des modules.

1. Vue Faculty
   1. Dashboard
   2. Courses
   3. Module
   4. Exam
   5. Profile
2. Vue Student
   1. Dashboard
   2. Courses
   3. Module
   4. Exam
   5. Profile