## <u>Final Assignment – Virtual Global College</u>

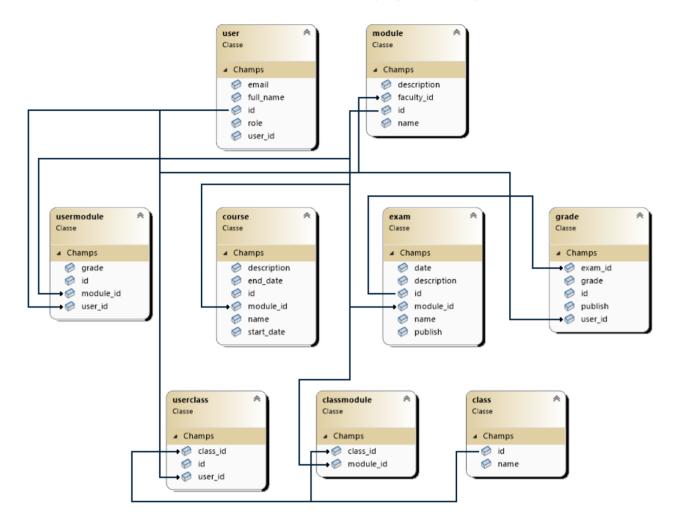
## Lucile Pelou

## Student n°74526

**Git repository:** https://github.com/Clarisse78/PooFinal

Video link: https://www.youtube.com/watch?v=ghb WxYA-Yk

For this assignment we could use any programming language and any method to complete the exercise. Having already manipulated Firebase with Node.js and JavaScript, I turned to this method. So, I first created a Firebase project with my different collections:



I created 9 collections as shown in the diagram above. To manage users and their authentications I used Firebase authentication which does this very well. The user\_id field in the user collection therefore refers to the user id in authentication and the password and email are managed by authentication. I added the email field in the user collection to make it easier.

After doing that I started my project by creating a git and my first page: login. This allowed me to see if the authentication and the user collection were working. Initially I created a global.js file where I put all the collections as well as the field names to be able to call them and change them more easily if necessary. I also did the password recovery method in case the user forgot it, Firebase Authentication already has a method in place to do this.

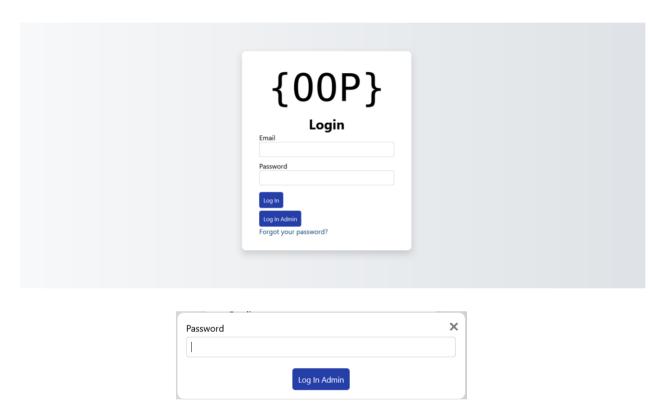
As collections are called with a string that does not change dynamically with the database, it was important that this was implemented for ease of use of the different collections.

After doing that I started building pages for a person with the admin role. To do this I created an admin folder and an admin\_dashboard page where I started to set up the vision logic of the different modules and users. I later realized that if I made one page per role it would quickly become a lot and that I would have to have lots of files with sometimes the

same methods or an overall file that was too large and not readable, which is not very ingenious. So, I changed my perspective and decided to make a dashboard page for everyone and that the content would change dynamically depending on the user's role. This approach allows for a much tidier workspace. I applied this to all the other pages I made afterwards. To be able to connect the given database with my website. I used all the methods such as getDoc, updateDoc, etc. imported from Firebase.

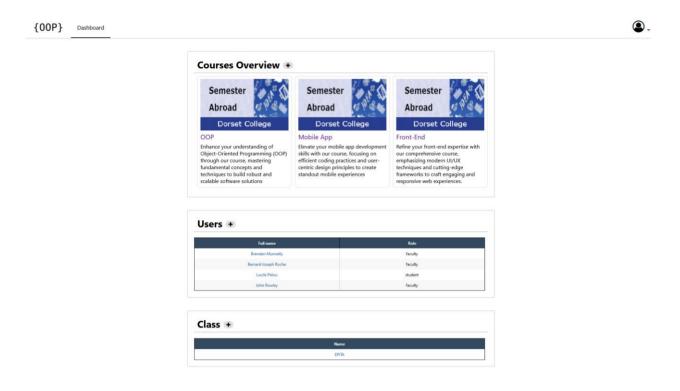
## 1) Login Page

First of all, if the user is not logged in, they end up on the login page. If he decides to connect as admin a dialog opens in which he will have to enter the password.

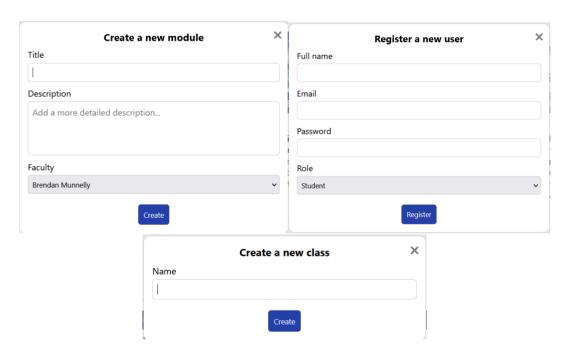


### 2) Admin view

#### a. Dashboard



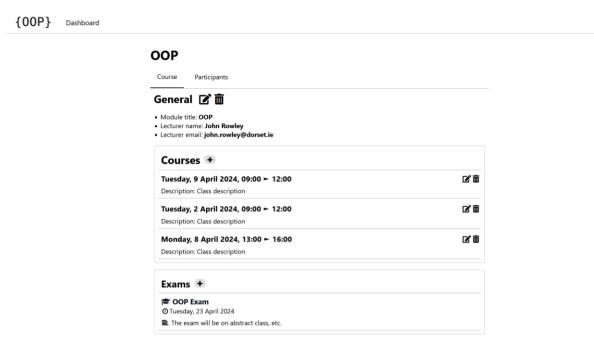
On this page the administrator can see all the modules as well as the list of users and classes. He can also add modules, users and classes by clicking on the plus buttons:



These buttons open dialogs allowing the administrator to enter information to create these new objects.

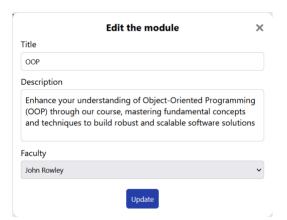
He can also click on the different objects to access their page and see their information.

#### b. Module

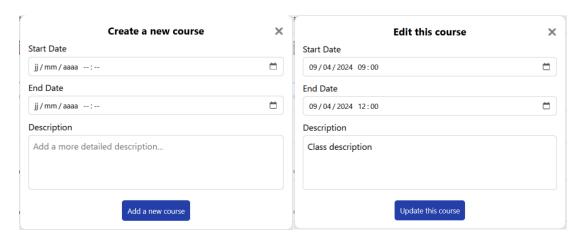


**(2**) ..

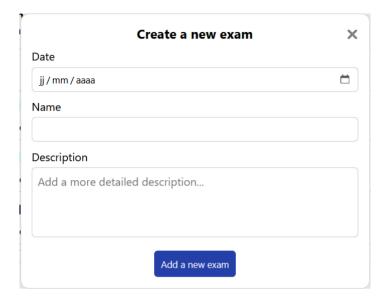
On this page, the administrator can see the information and objects attached to the module. There are two pages, course and participants. In the first there is first of all the information of the module such as the title, the name of the professor as well as his email. Below there are the courses as well as the exams for the module. At the very top, next to general it is possible to click on the edit icon to be able to edit the module:



The trash allows him to delete it (along with all the objects attached to it). In the courses he can add a new one, edit and delete those already present.

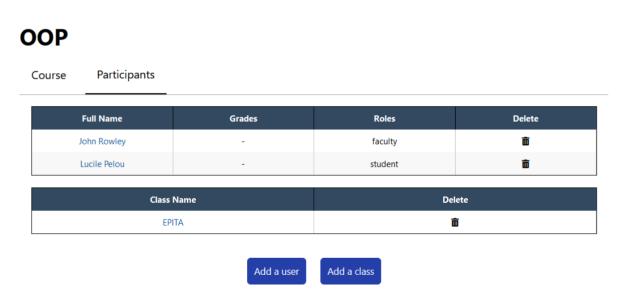


For exams, you can create new ones by clicking on the plus button:



And he can also see the information of those already created by clicking on their titles.

The participant page allows him to see all the users linked to this course as well as their grade. He can add and remove some. He cannot edit the mark since it is the combination of all the exam marks. He can also see the classes and just like the users he can add or remove them from the module.

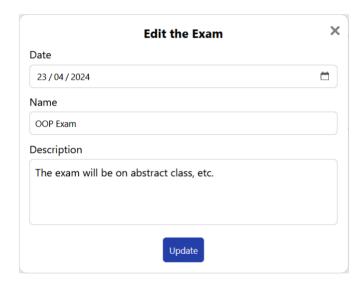


#### c. Exam

As said previously, the admin can click on an exam title in the module page to see its information.



Here he can see information such as the name of the module, the title of the exam, its date and its description. He can edit and delete it.



He can access the Grades page which will show him the exam participants and their grades.

## **OOP/OOP Exam**

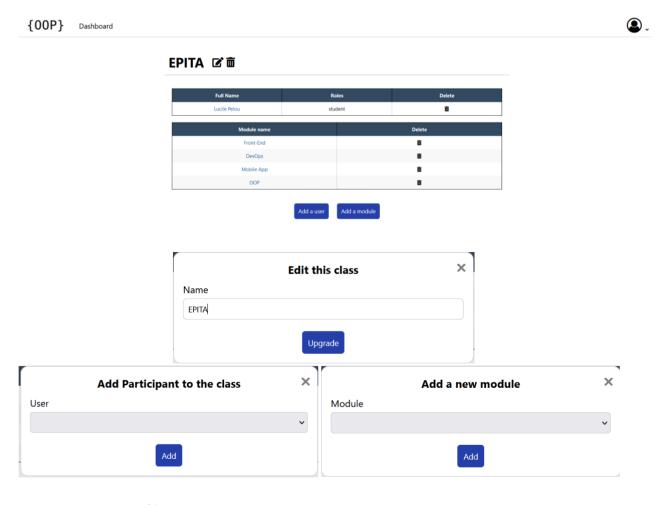


	Edit this Grade	×
Grade		
		\$
Publish 🗆		
	Update	

He can edit the grades and decide whether to publish them or not. The publish all grades button allows you to publish or unpublish them all at once

## d. Class

The class page allows the admin to see the students and modules attached to it. He can also modify the class, delete it or add students or modules.



e. Profile

The admin doesn't have a profile since he doesn't have any courses so there's no point. On the other hand, he can go and see the user profile, for example by clicking on the first name on the table in one of the previous image he can access his profile.



**②** ,

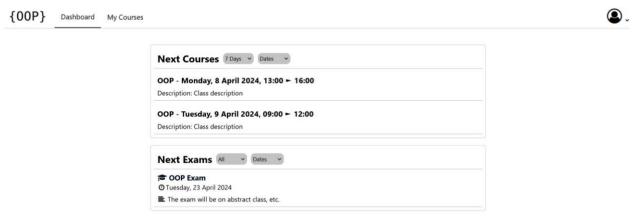


From here he can see the user's information and modify it. He can also add or remove modules and classes.

### 3) Faculty View

#### a. Dashboard

From the dashboard, the teacher can see the next courses as well as the next exams. He can access the exams by clicking on them.



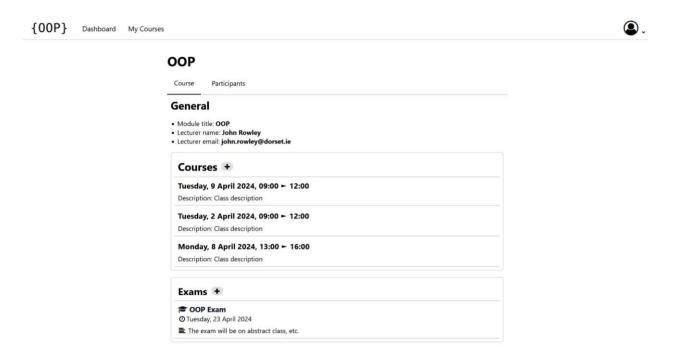
#### b. Courses

The courses page will allow him to see all his courses and be able to access them  $\{00P\}$  Dashboard My Courses



#### c. Module

The module page is almost identical to that of the admins.



The only difference is that he cannot edit the module or delete it. On the other hand, he can add courses or exams.

For the participants page it is also almost identical but he cannot add users or classes.



#### d. Exam

It's exactly the same as for administrators.

### e. Class

He can access the classes page but cannot modify them.

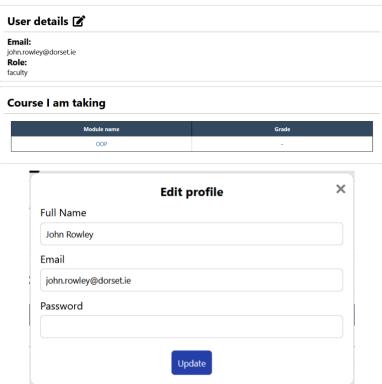
## **EPITA**



### f. Profile

He has access to his profile and can modify his information such as his email or password. He can also see the profile page of other users but cannot modify anything.

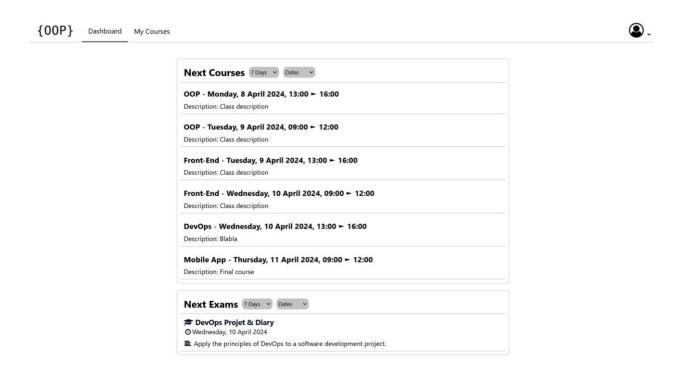
# John Rowley



## 4) Student view

#### a. Dashboard

Just like the faculty view, the student sees their upcoming courses and exams.

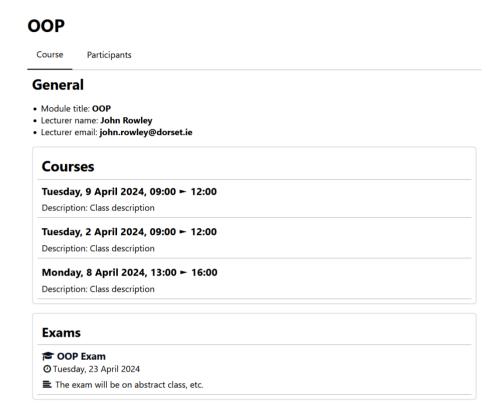


#### b. Courses

Like the dashboard, this page is the same as for the faculty view.

#### c. Module

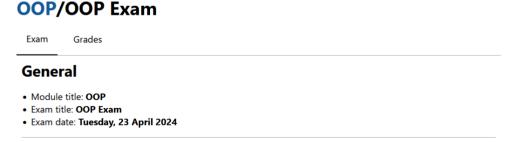
Once again, the page is almost identical to previous views. The difference is that he cannot modify anything.



The participant page is exactly the same as with the teacher view.

### d. Exam

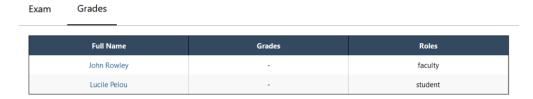
The exam page shows the exam information and the user cannot edit it or delete the exam.



The exam will be on abstract class, etc.

For the exam grades page, the user will not see the grades if they have not yet been published, unlike the admin and teachers.

## **OOP/OOP Exam**



## e. Class

This page is the same as on the teacher view.

## f. Profile

Just like teachers, students can modify their information. They have access to the professors' profiles, not those of other students.