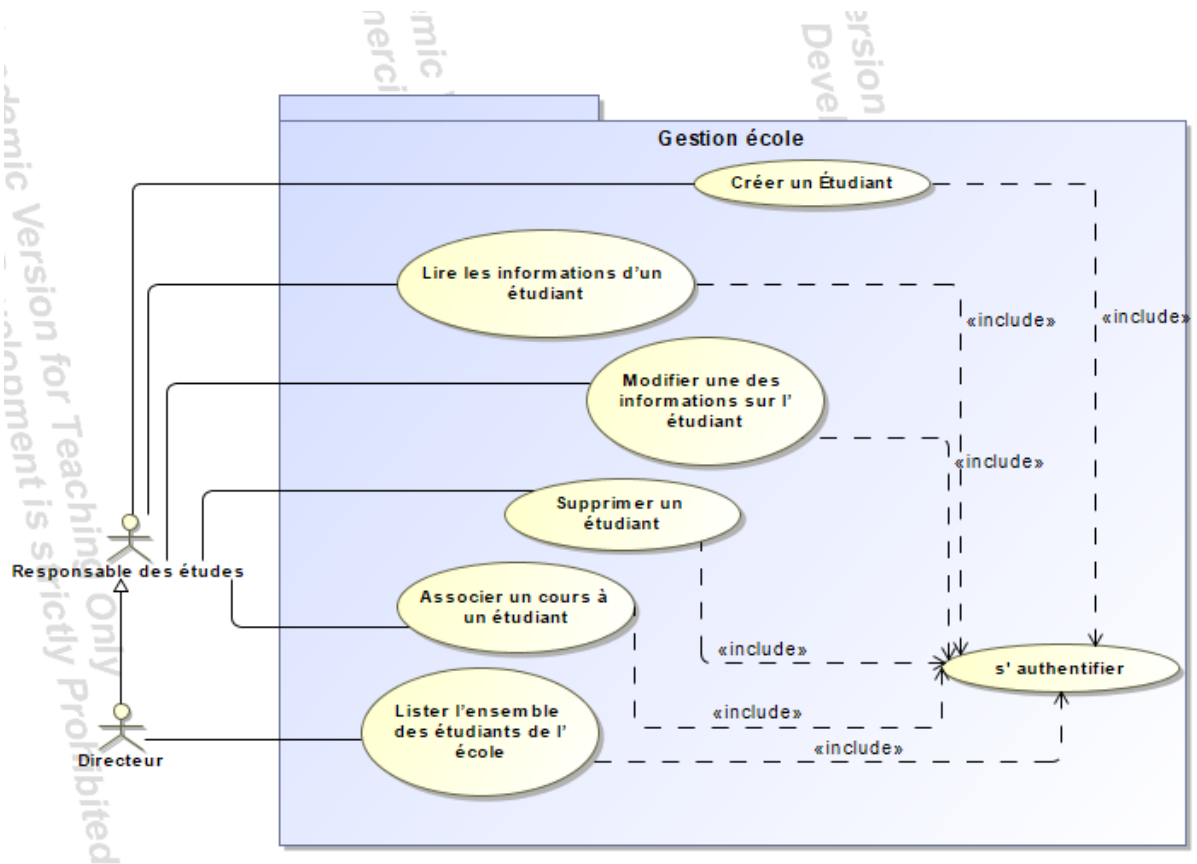


Projet client 16/03/2022

Projet client 16/03/2022	0
Diagrammes	6
Use Case	6
Activité	6
S'authentifier	7
Créer un étudiant	7
Modifier un étudiant	8
Supprimer un étudiant	9
Lire les informations d'un étudiant	10
Associer un cour à un étudiant	11
Lister tous les étudiants	14
Classe	14
Séquence	15
S'authentifier	15
Créer un étudiant	15
Modifier un étudiant	16
Supprimer un étudiant	16
Associer un cours à un étudiant	16
Lister les étudiants	16
Développement	16
Technologies utilisés	16
Back	16
Front	16
Ma base de données avec des données	16
Résultat des tests	17
Devops	18
Jenkins	18
Docker	20
Ansible	20
Terraform	20
Résultat sur le cloud AWS	20
Liens	20
Docker image	20
Git hub	21
projet	21
Déploiement	
https://github.com/Flav1-ann/course_deploy	0
AWS	0

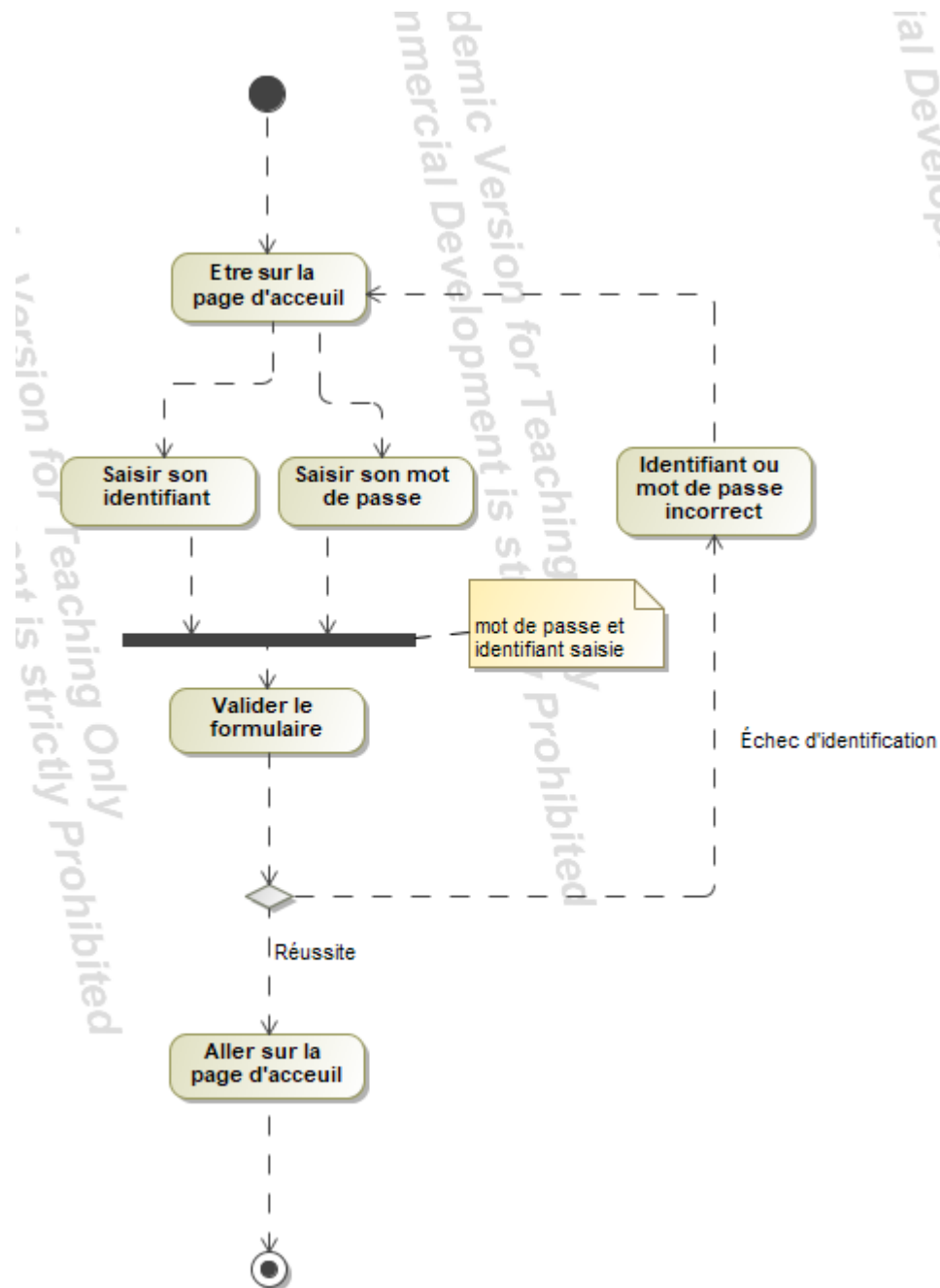
Diagrammes

Use Case

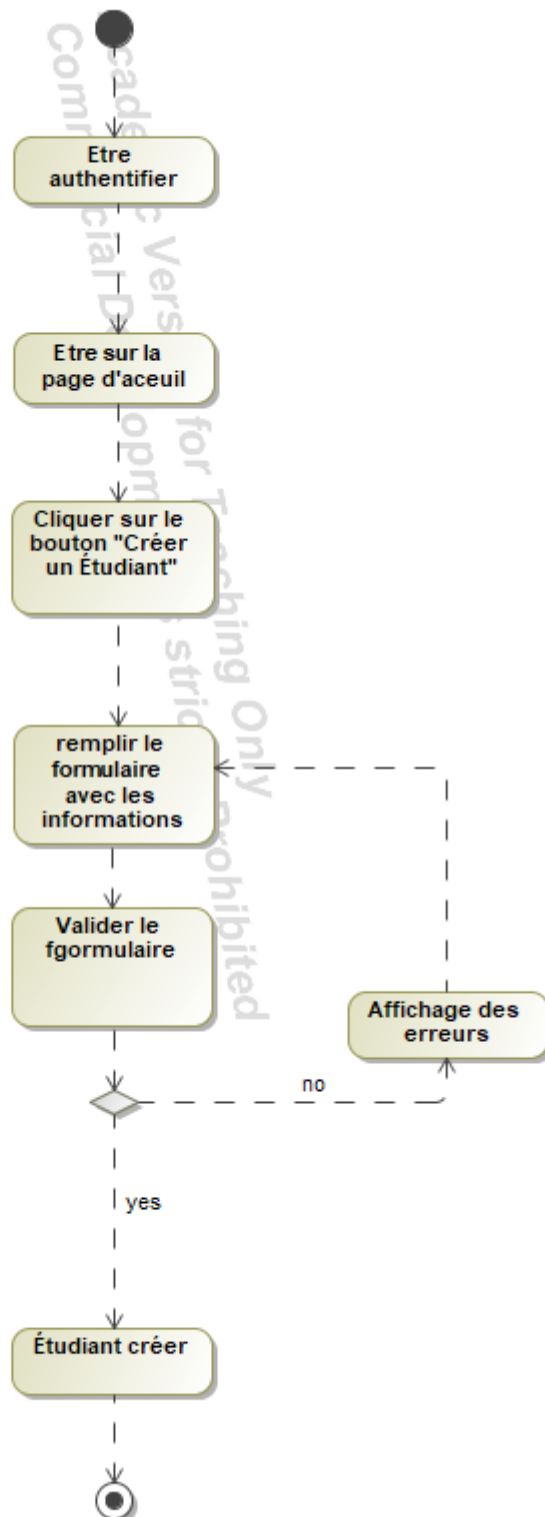


Activité

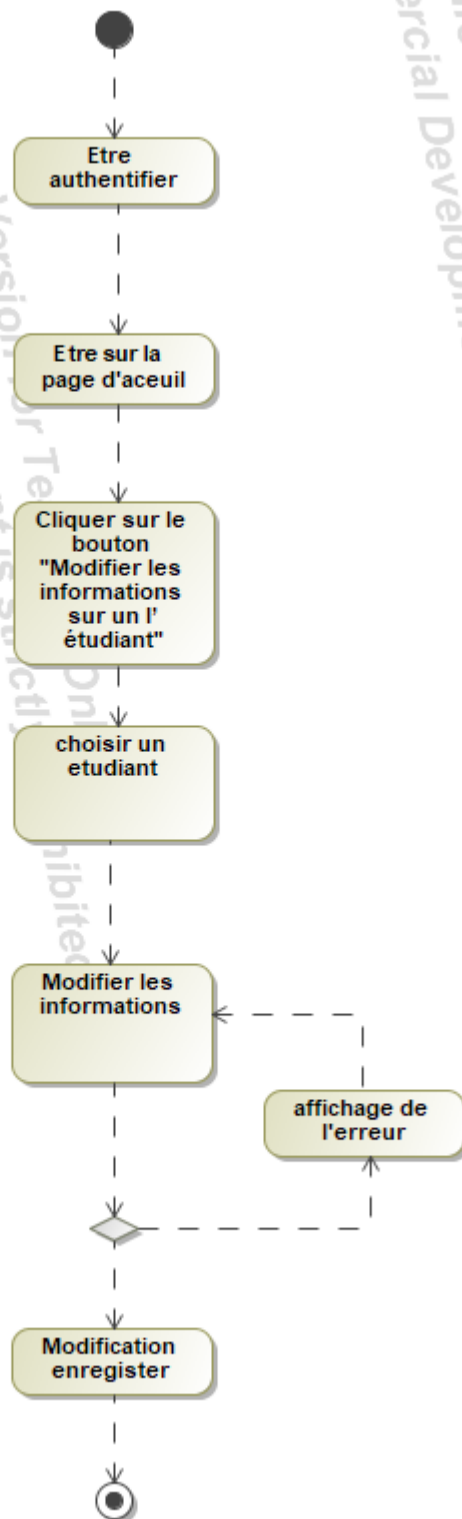
S'autentifier



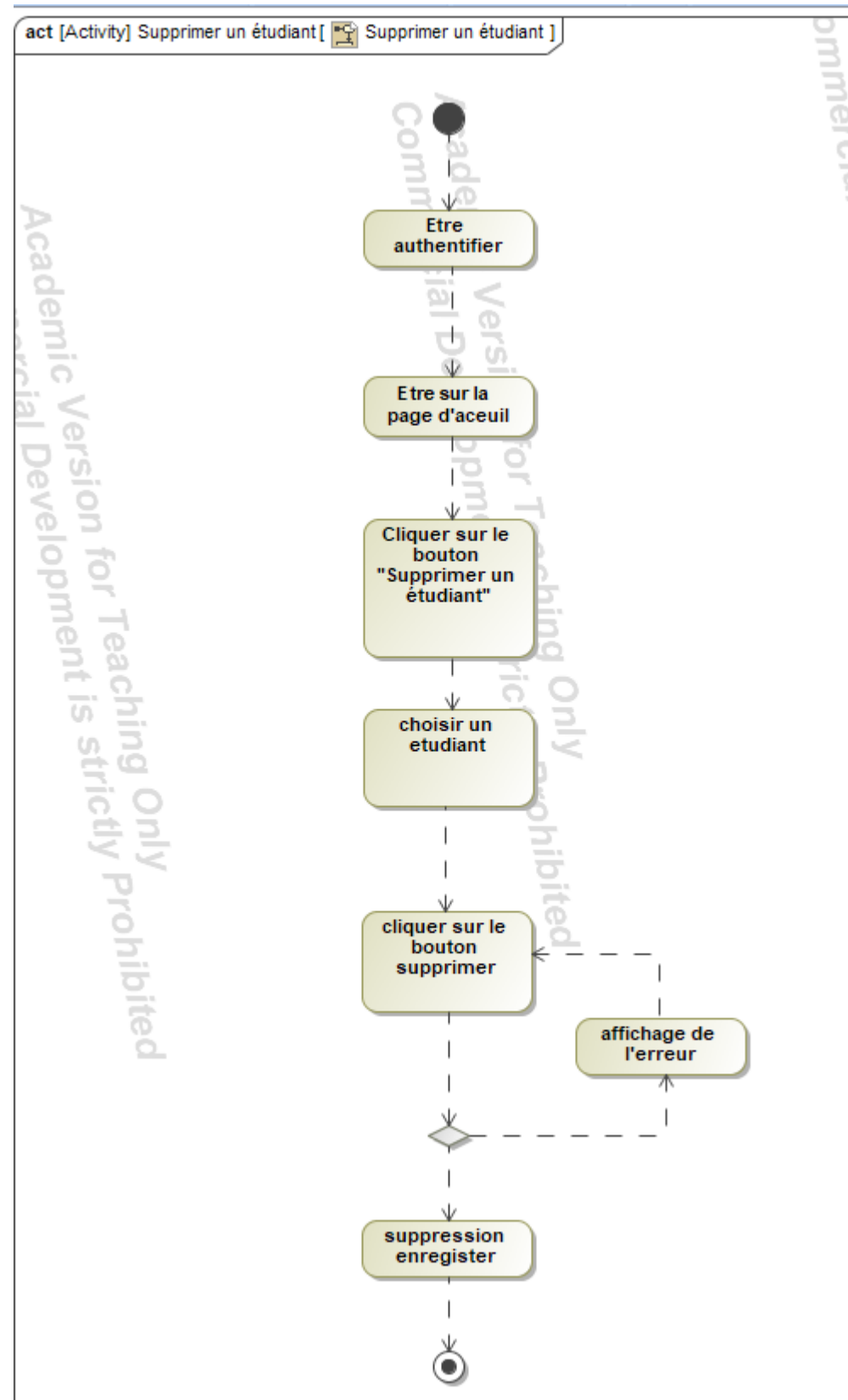
Créer un étudiant



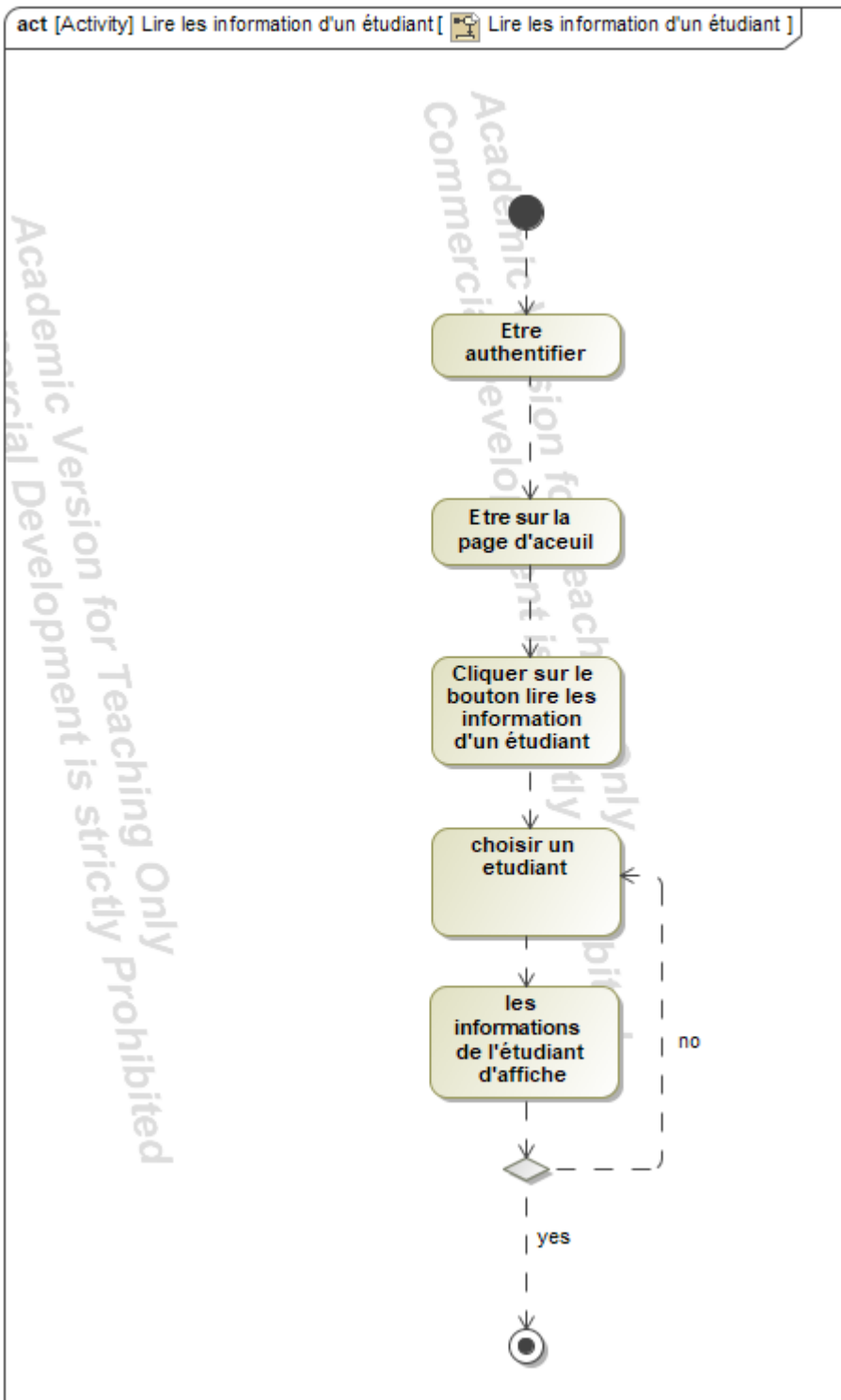
Modifier un étudiant



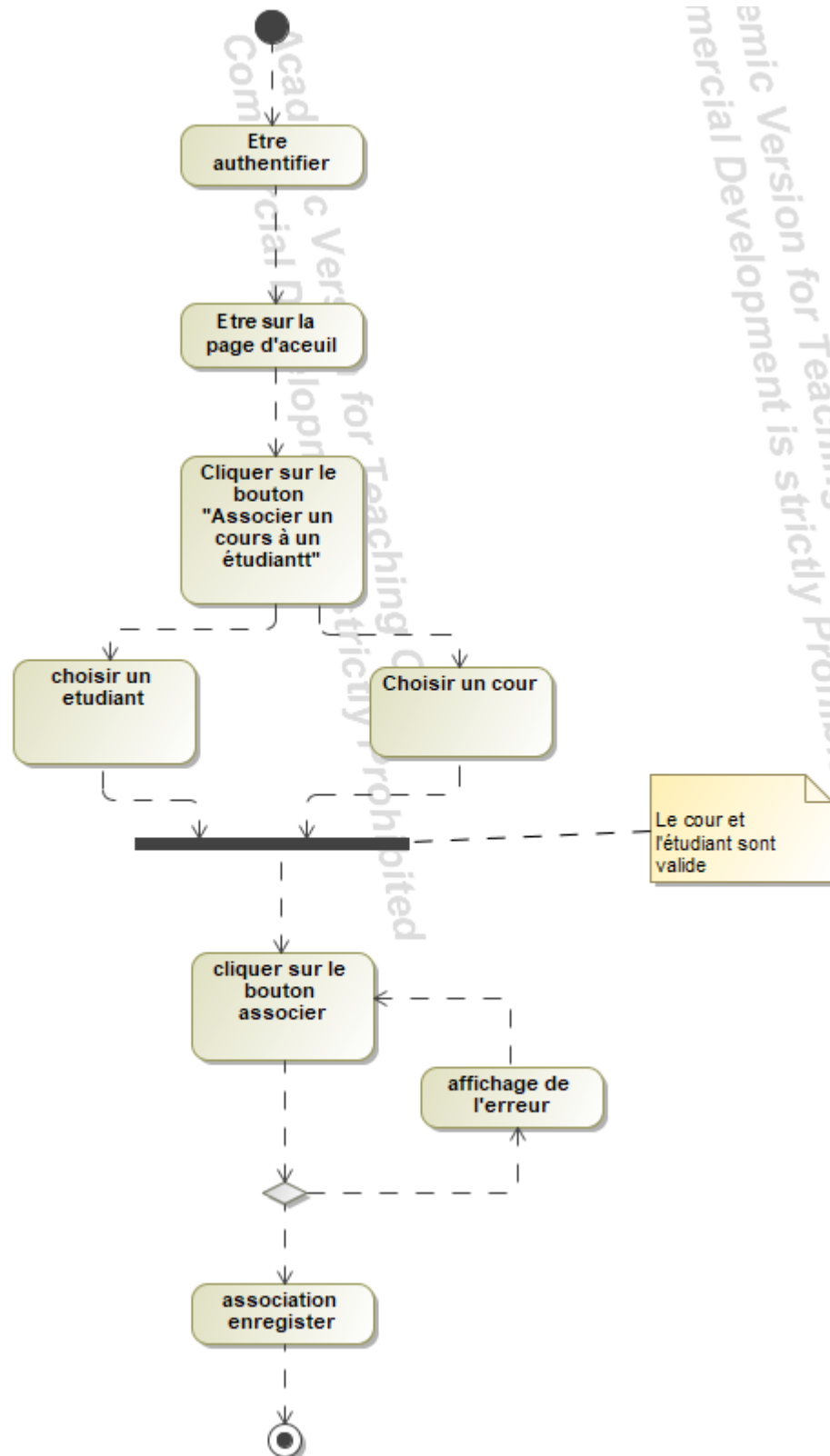
Supprimer un étudiant



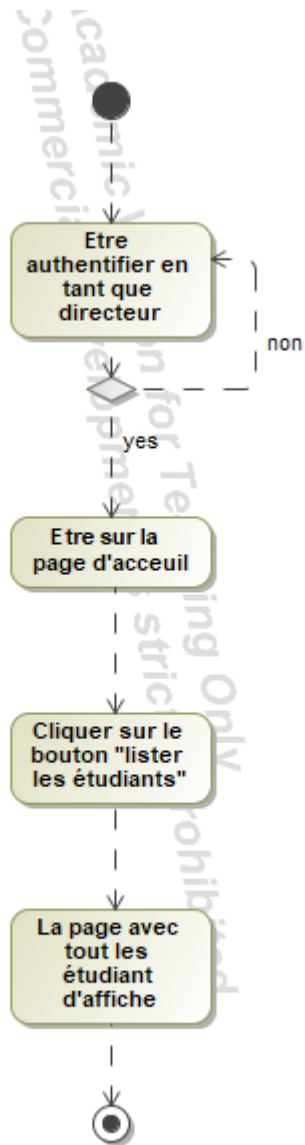
Lire les informations d'un étudiant



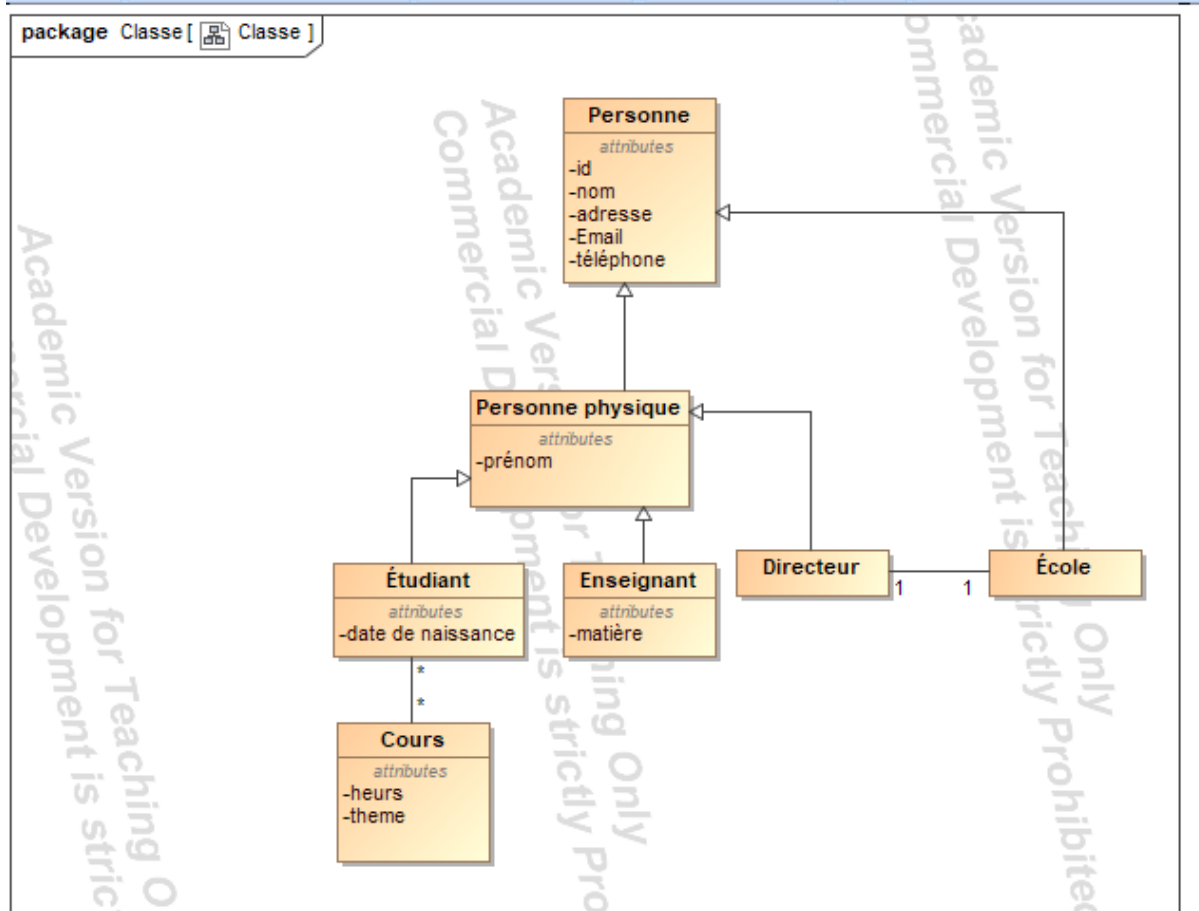
Associer un cour à un étudiant



Lister tous les étudiants

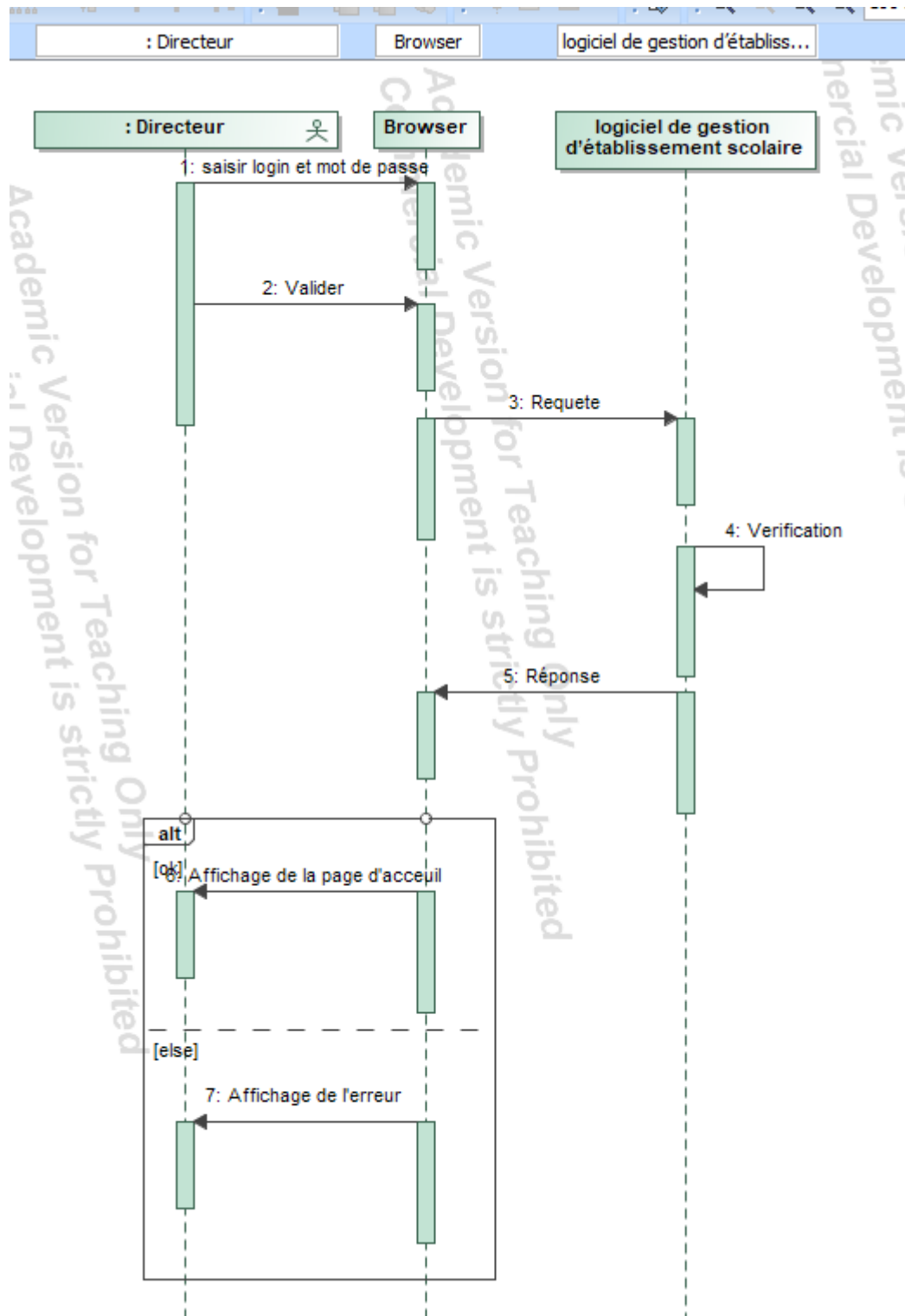


Classe

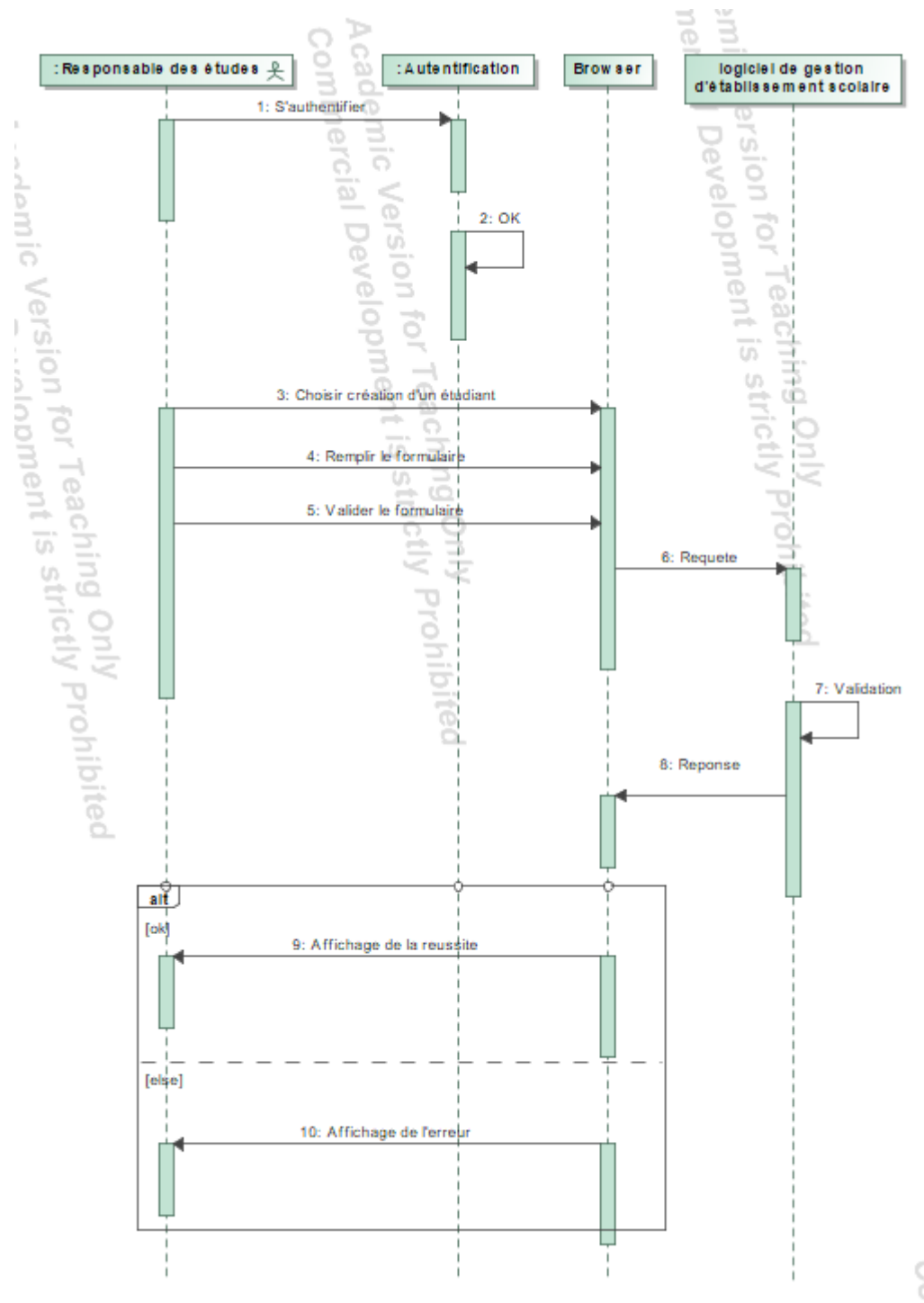


Séquence

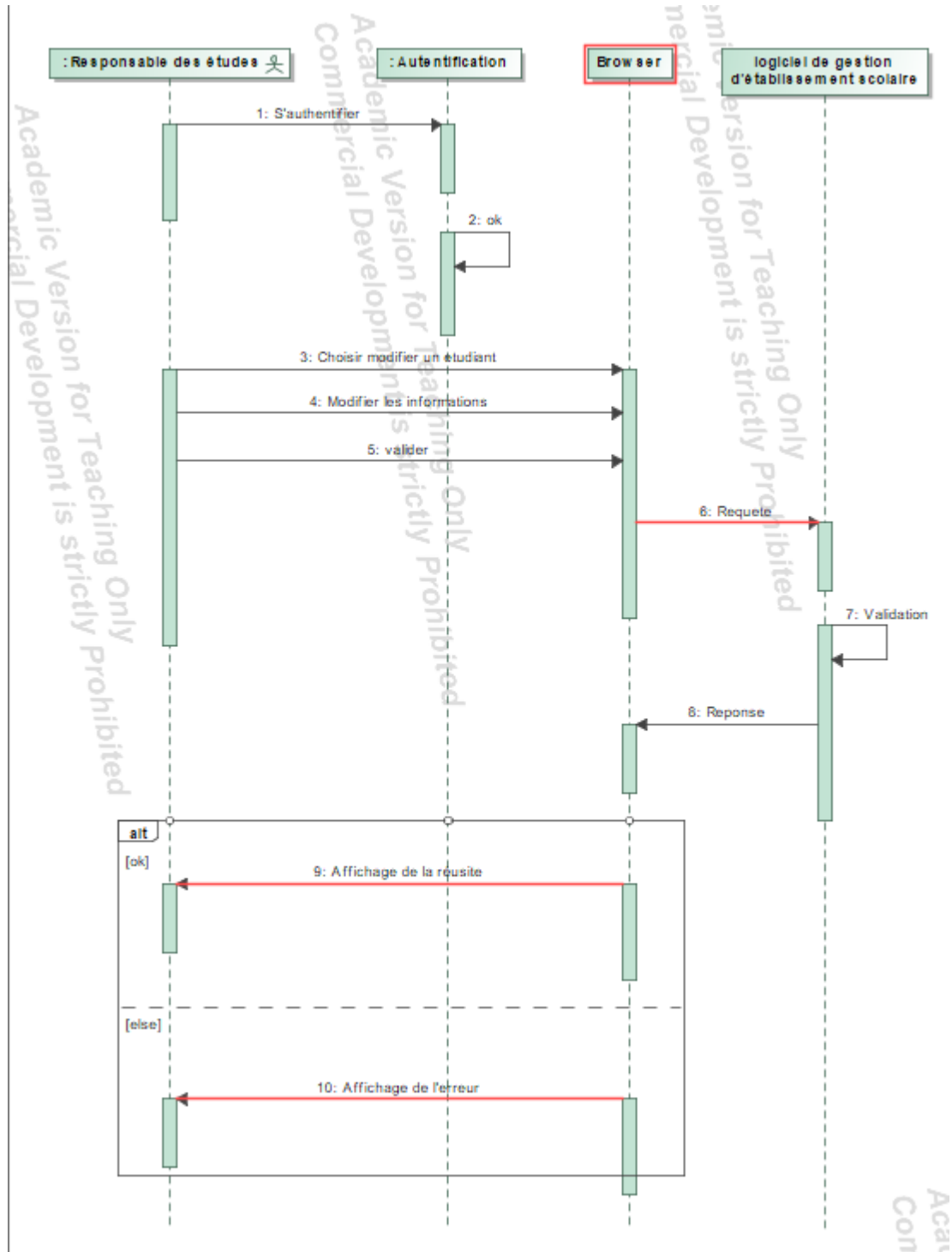
S'authentifier



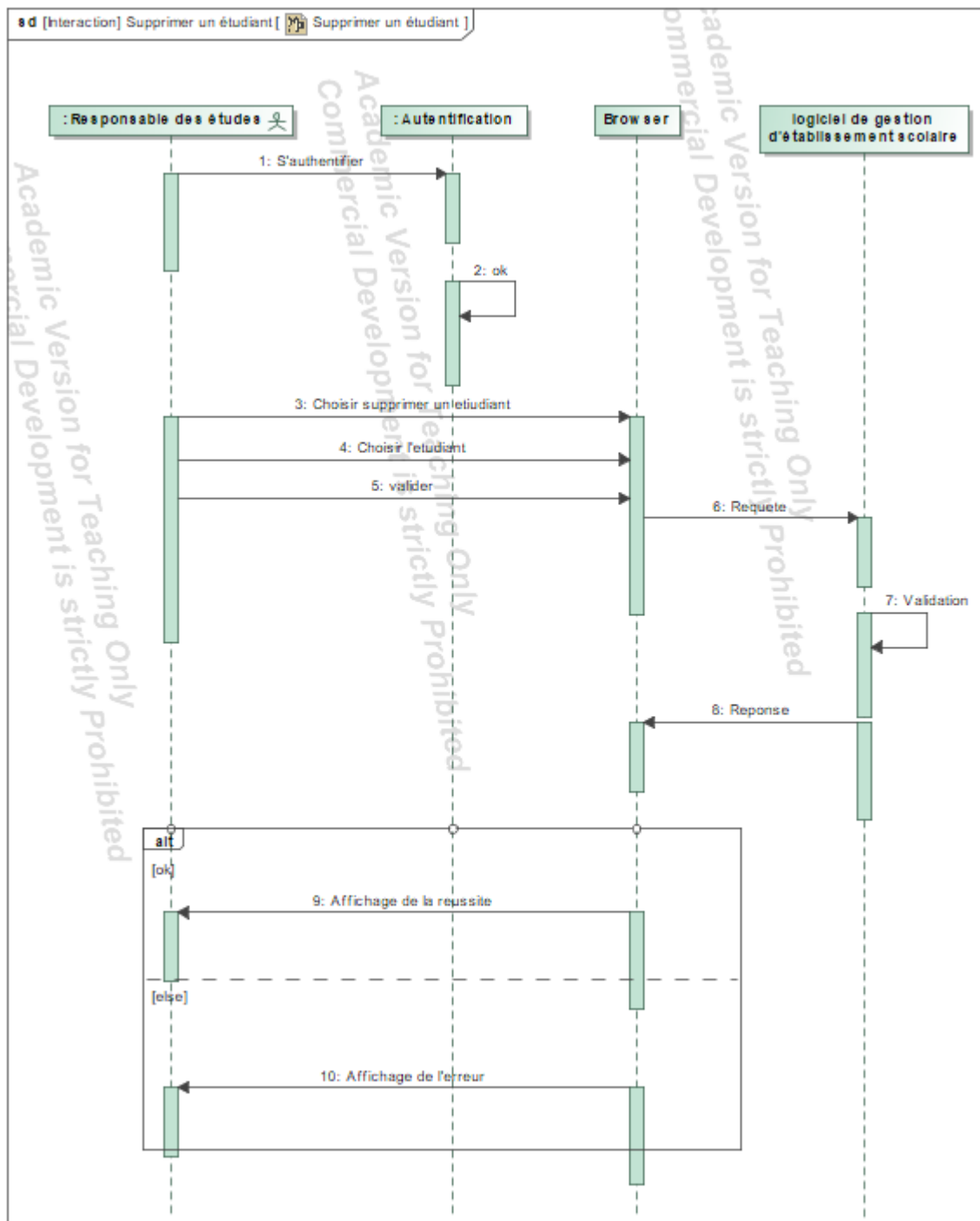
Créer un étudiant



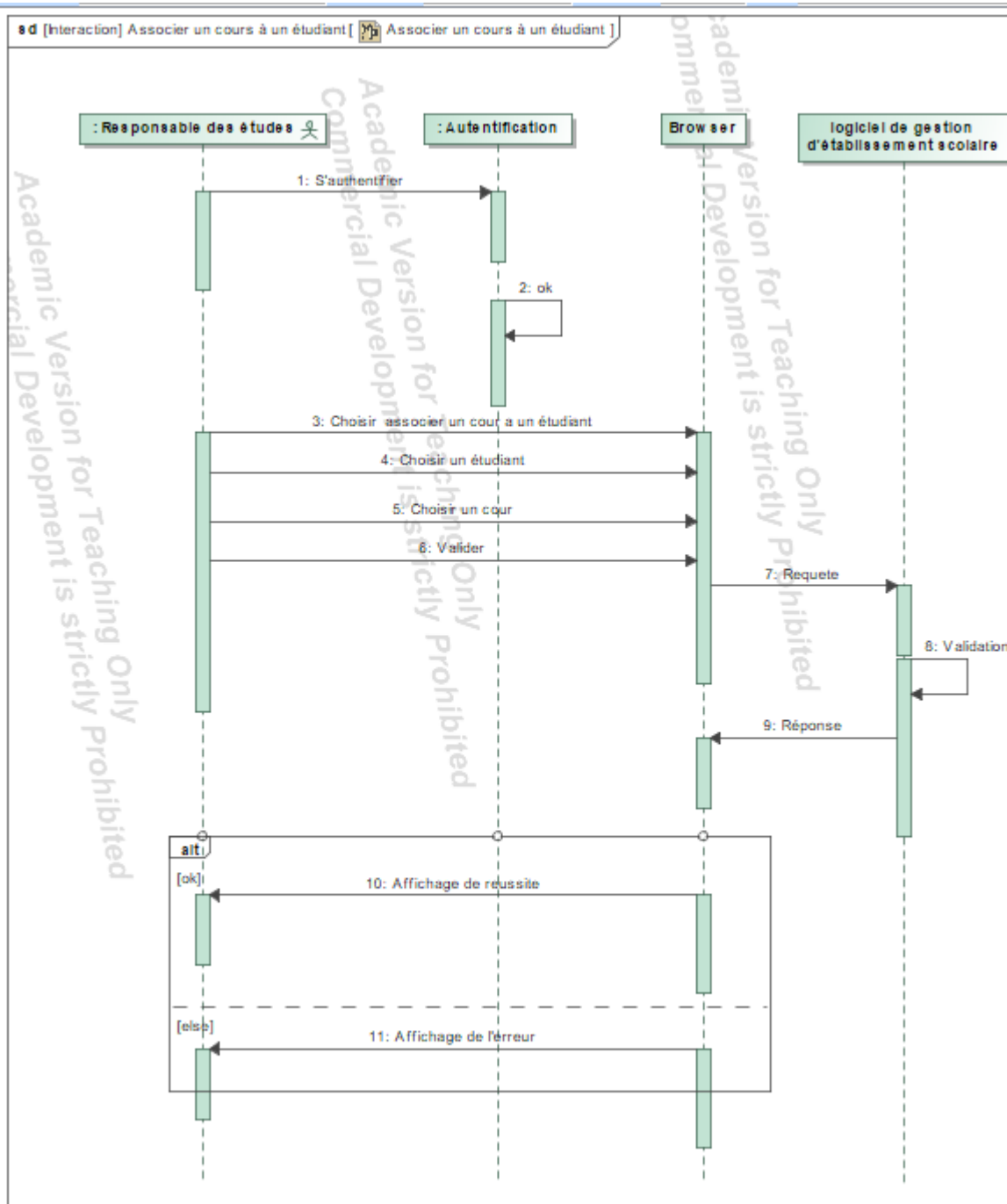
Modifier un étudiant



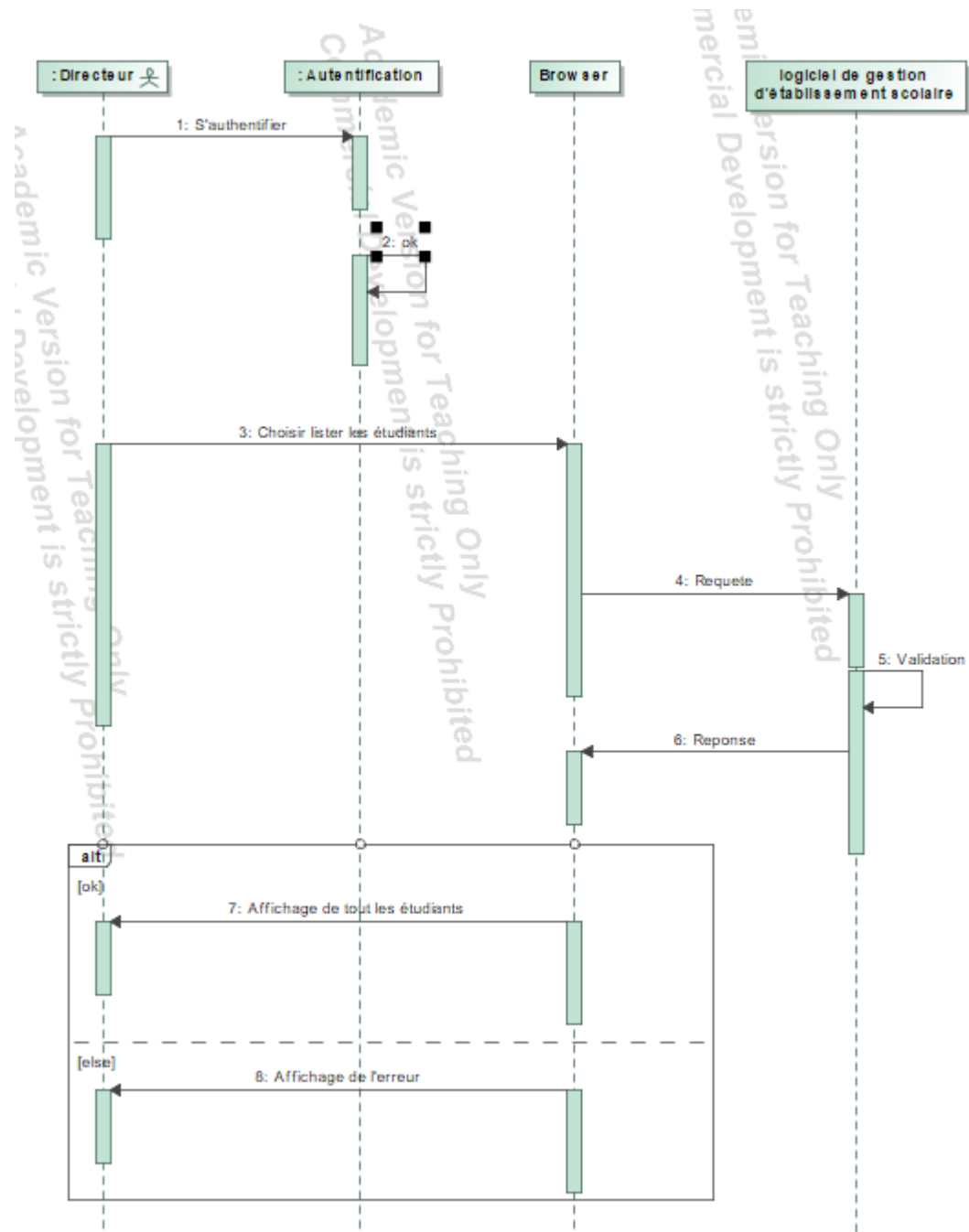
Supprimer un étudiant



Associer un cours à un étudiant



Lister les étudiants



Développement

Flavien ANNAIX
Master 2

Technologies utilisés

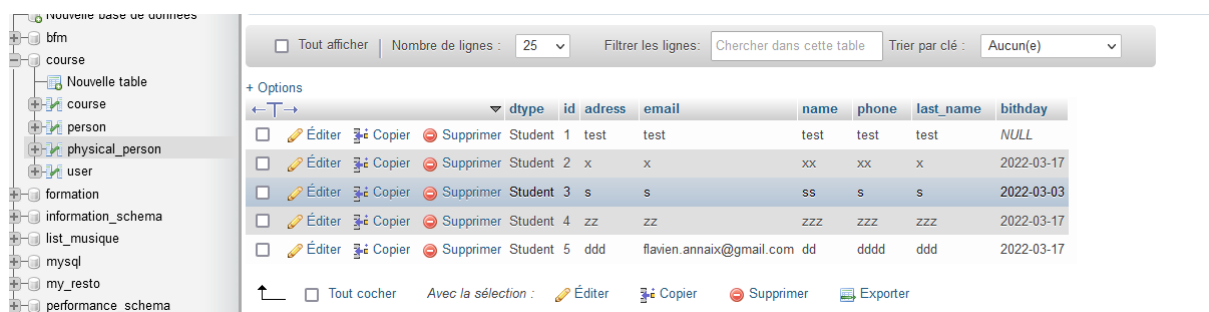
Back

Pour le back de l'application j'ai utilisé Spring Boot avec jacoco pour le code coverage et Junit5 pour les tests

Front

Pour le Front j'ai utiliser le HTML5, CSS, BootStrap

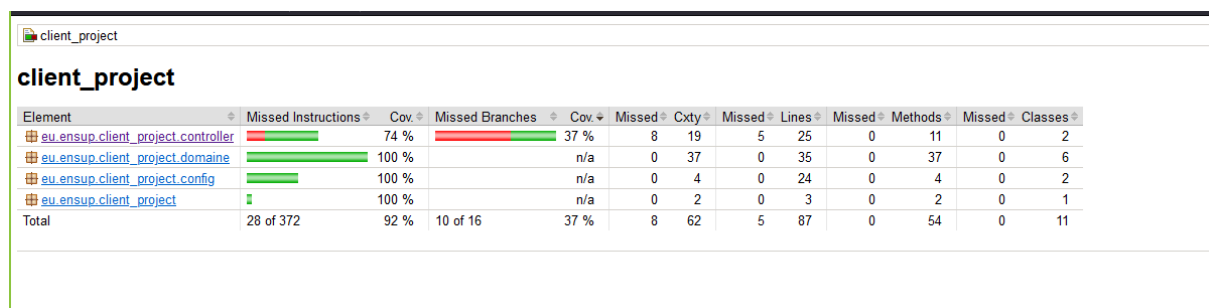
Ma base de données avec des données



The screenshot shows a database management interface. On the left is a tree view of the database structure. The main area displays a table with the following data:

	dtype	id	adress	email	name	phone	last_name	bithday
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Student	1	test	test	test	test	test	NULL
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Student	2	x	x	xx	xx	x	2022-03-17
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Student	3	s	s	ss	s	s	2022-03-03
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Student	4	zz	zz	zzz	zzz	zzz	2022-03-17
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Student	5	ddd	flavien.annaix@gmail.com	dd	ddd	ddd	2022-03-17

Résultat des tests



The screenshot shows a code coverage report for the 'client_project'. The report includes a table with the following data:

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
eu.ensup.client.project.controller	74 %	37 %	8	19	5	25	0	11
eu.ensup.client.project.domaine	100 %	n/a	0	37	0	35	0	37
eu.ensup.client.project.config	100 %	n/a	0	4	0	24	0	4
eu.ensup.client.project	100 %	n/a	0	2	0	3	0	2
Total	28 of 372	92 %	10 of 16	37 %	8	62	5	87

dans le repo [github](#) dans le dossier testCoverage il y a le rapport jacoco

Devops

Jenkins

NON TRAITÉ

Docker

Le dockerFile

```
FROM openjdk:15-jdk-alpine

LABEL MAINTENER="Flavien ANNAIX"

RUN apk update && \
    apk upgrade && \
    apk add git && \
    apk add maven && \
    apk add bash

RUN git clone https://github.com/Flav1-ann/course /course

WORKDIR /course

RUN mvn clean package -DskipTests=true

RUN cp target/*.jar app.jar

#RUN addgroup -S spring && adduser -S spring -G spring

EXPOSE 80

#USER spring:spring

ENTRYPOINT ["java", "-jar", "app.jar"]
```

Docker Compose

```
m.tfvars x flavi1-ann-course.2022-03-16.private-key.pem x docker-compose.yml x
version: '3.4'
services:
  server:
    image: flavi1ann/course
    restart: always
    depends_on:
      - db
    network_mode: host
  db:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    cap_add:
      - SYS_NICE
    environment:
      - MYSQL_USER=${MYSQL_USER}
      - MYSQL_PASSWORD=${MYSQL_PASSWORD}
      - MYSQL_DATABASE=${MYSQL_DATABASE}
      - MYSQL_ALLOW_EMPTY_PASSWORD=no
    ports:
      - 3306:3306
```

Ansible

playbook.yml

```
---
- name: playbook
  hosts: webserver
  vars_files:
    - private_settings.yml
  environment:
    MYSQL_USER: 'spring'
    MYSQL_PASSWORD: '{{ MYSQL_PASSWORD }}'
    MYSQL_DATABASE: 'course'

  pre_tasks:
    - name: Install aptitude using apt
      apt: name=aptitude update_cache=yes force_apt_get=yes
        become: yes

    - name: Install required system packages
      apt: name={{ item }} update_cache=yes
        become: yes
      loop:
        [
          "apt-transport-https",
          "ca-certificates",
          "curl",
          "software-properties-common",
          "python3-pip",
          "python-apt",
          "virtualenv",
          "python-setuptools",
          "net-tools",
          "apache2"
        ]

    - name: Enable apache2
      service:
```

```
-
- name: Add Docker Repository
  become: yes
  apt_repository:
    repo: deb https://download.docker.com/linux/ubuntu bionic stable
    state: present

- name: Update apt and install docker-ce and docker-compose
  become: yes
  apt: name={{ item }} update_cache=yes
  loop: [ "docker-ce", "docker-compose" ]

- name: Install Python Docker Module
  pip:
    name: ["docker", "docker-compose==1.25.0"]

- name: Copy to remote
  copy:
    src: ./docker_usr_config.sh
    dest: '{{ WORKING_DIR }}'
    mode: u=rxw,g=rw,o=r

- name: Add user to docker group
  become: true
  shell:
    cmd: ./docker_usr_config.sh &
  args:
    chdir: '{{ WORKING_DIR }}'

tasks:
- name: Copy to remote
  copy:
    src: ./docker-compose.yml
    dest: '{{ WORKING_DIR }}'

- name: run docker-compose
  become: true
  community.docker.docker_compose:
    project_src: '{{ WORKING_DIR }}'
```

Terraform

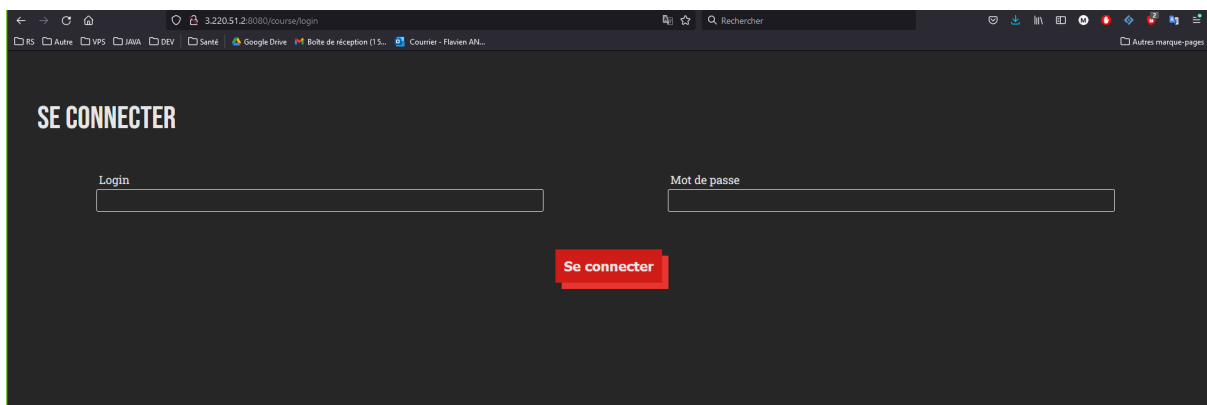
Voici le main mais tous les fichiers sont ici

https://github.com/Flav1-ann/course_deploy/tree/master/app et

https://github.com/Flav1-ann/course_deploy/tree/master/Modules

```
1  module "ec2" {
2      source          = "../Modules/EC2"
3      author_name     = var.author_name
4      instance_type   = var.instance_type
5      private_key_path = var.private_key_path
6      availability_zone = var.ec2_avail_zone
7      sg_name         = module.sg.out-sg-name
8      public_ip       = module.eip.out_eip_ip
9      main_directory  = var.main_dir
10 }
11
12 module "sg" {
13     source = "../Modules/SG"
14     tag_name = var.author_name
15 }
16
17 module "eip" {
18     source      = "../Modules/EIP"
19     author_name = var.author_name
20 }
21
22 resource "aws_eip_association" "eip_association" {
23     allocation_id = module.eip.out_eip_id
24     instance_id   = module.ec2.out-ec2-id
25 }
```

Résultat sur le cloud AWS



The screenshot shows a web browser window with a dark theme. The address bar shows the URL '3.220.51.2.8080/course/login'. The page has a header with navigation links: 'RS', 'Autre', 'VPS', 'IWA', 'DEV', 'Santé', 'Google Drive', 'Boîte de réception (15)', and 'Courrier - Flavien AN...'. The main content area is titled 'SE CONNECTER' and contains two input fields: 'Login' and 'Mot de passe'. Below these fields is a red button labeled 'Se connecter'.

Liens

Docker image

<https://hub.docker.com/repository/docker/flav1ann/course>

Git hub

projet

<https://github.com/Flav1-ann/course>

Déploiement

https://github.com/Flav1-ann/course_deploy

AWS

<http://3.220.51.2:8080/course/login>