Project specification note

MOOC-project

Maria Kononevskaya

Paris 2021

Table of content

Summary

Functional specification

Objective of the project

The objective of this project is to create a website-hub to store courses of the various subjects, where users can learn new skills both for professional and leisure purposes. The website is open for anyone, but registered users have a possibility to both comment on courses and create courses of their own (depending on user type).

Users

There are three types of users:

- Unregistered user (visitor)

Visitors can browse and take courses as well as see comments and notes. They are also able to register.

- Registered user (lambda)

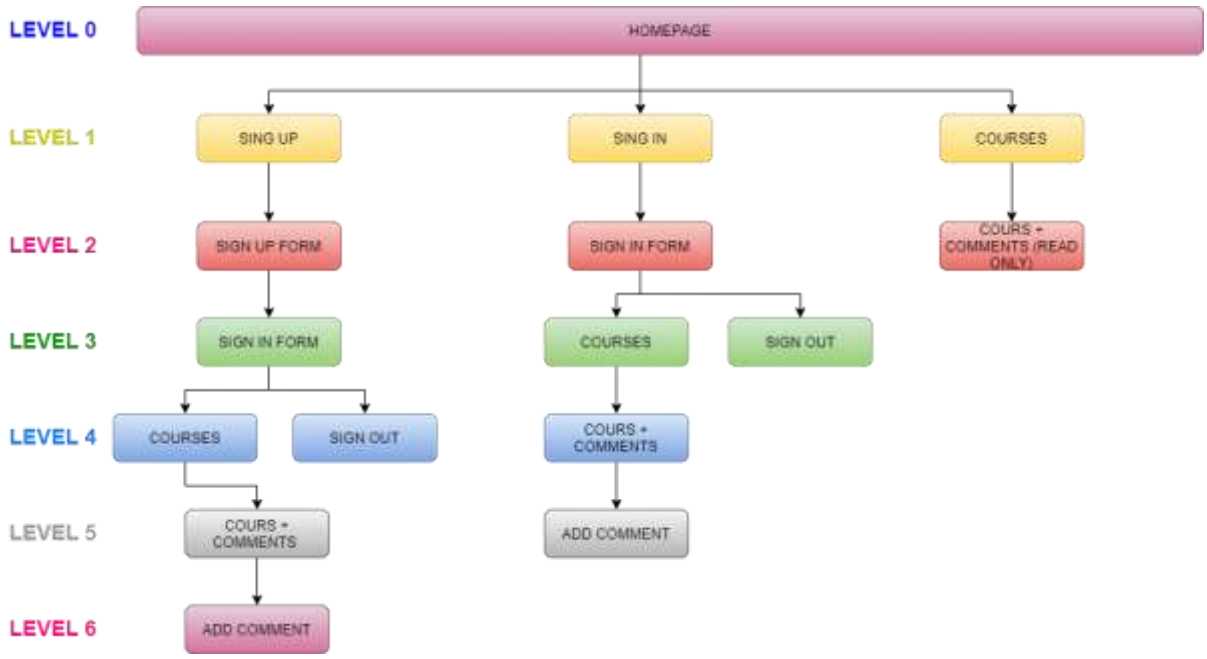
Lambda users can not only browse and take courses as well as see comments and notes, they are also authorized to add comments and review (= note) courses.

- Registered user (super)

Super users have the rights to do everything stated previously as well as upgrade other users, delete comments and update or delete courses.

Hierarchical structure of the website

This hierarchical structure does not include super user.



Our webpage consists of homepage with the list of courses and the possibility to either log in or register as well as add course for super users, login page, register page, course page with comment section and the possibility to add comments for registered (lambda and super) users.

Technical specification

There are many technologies which can be used to create this project. One option is to create the front with frameworks like React or Vue and deploy it to Azure via Azure Static Web Apps using Github and Github action's CI/CD. The API can be written with Azure functions (mode serverless).

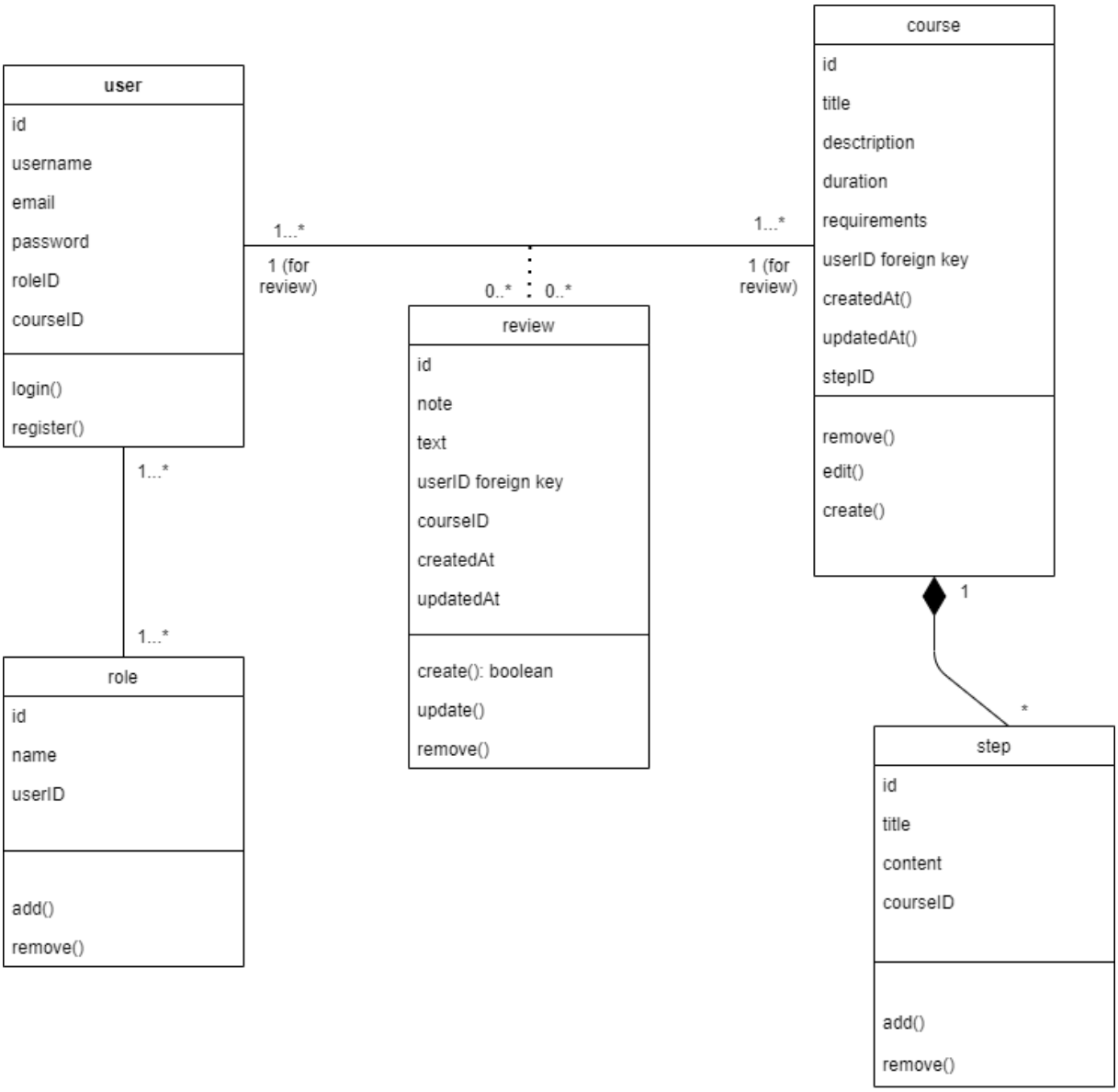
Another option is to use Adonisjs for API and for example Postgres or SQL for database. Adonis is using Lucid ORM which helps us create tables and perform queries.

Backend can be also created with Nodejs and Express, for example. We can use any Azure DB, managed instance or even some SQL server running on the Azure VM.

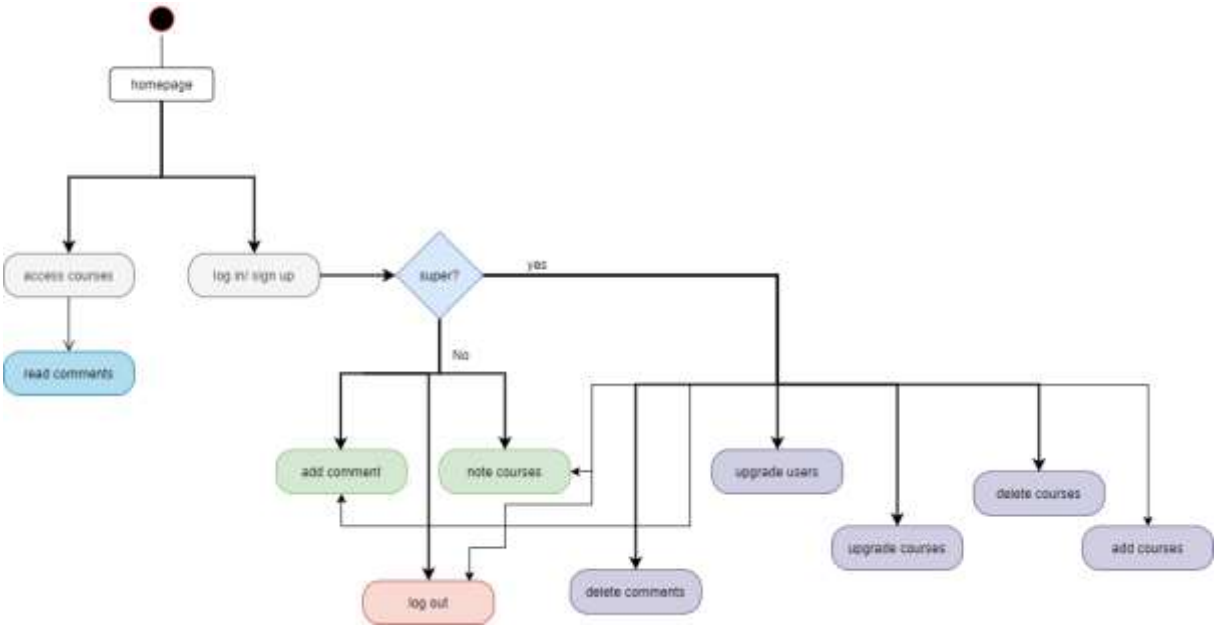
At this moment we've only decided to create the front with Vue.js. Backend option is still open, same as database, even though it is sure that we will be using Azure's relational database (either SQL or Postgres).

Diagrams

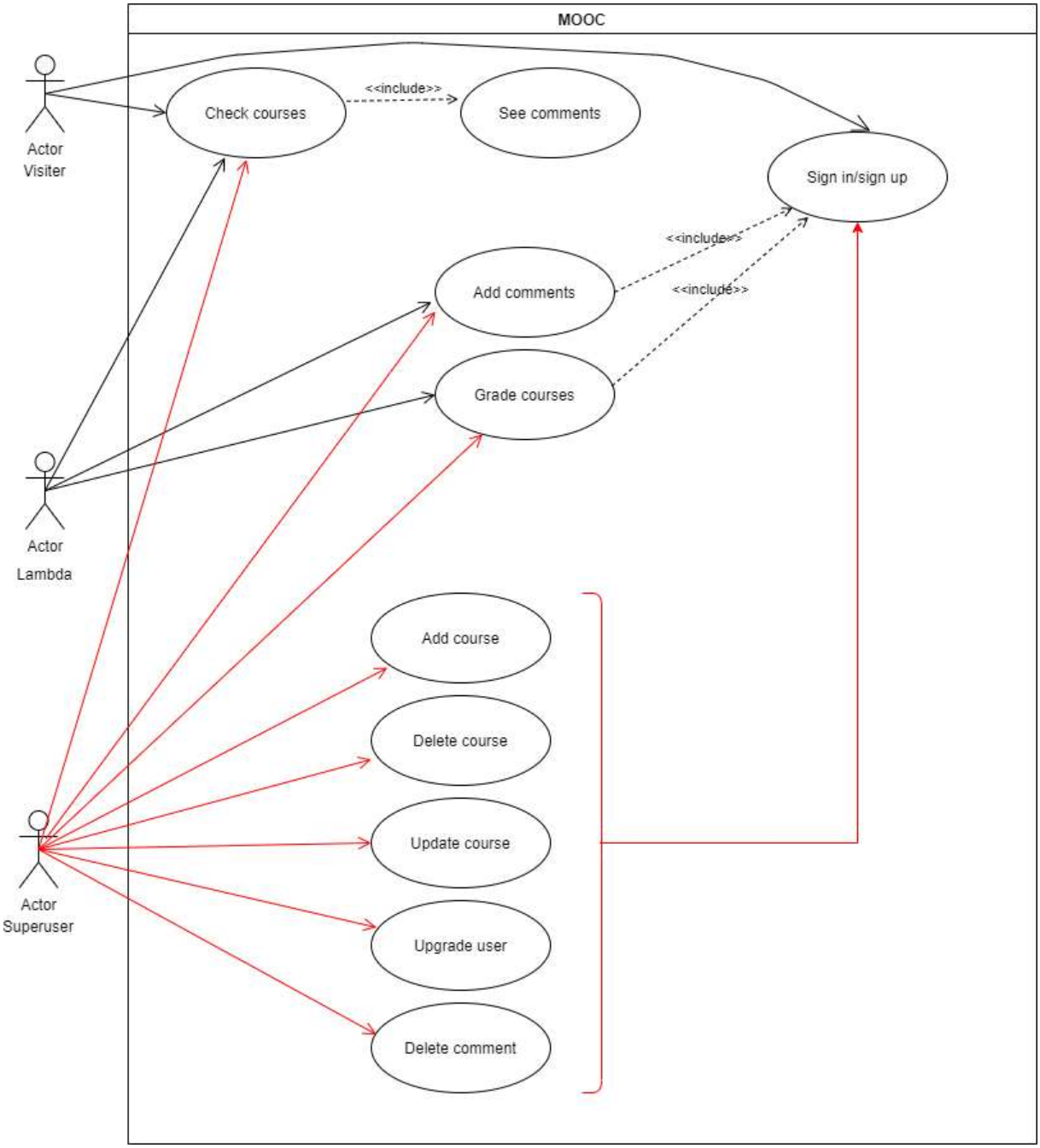
Class diagram



Activity diagram



Use case diagram



Physical data model