Software Engineering – 04GSP

Books or notes are **not** allowed.
Write only on these sheets. **Concise** and **readable** answers please.

Surname, name, matricola ____

Class attendance in university

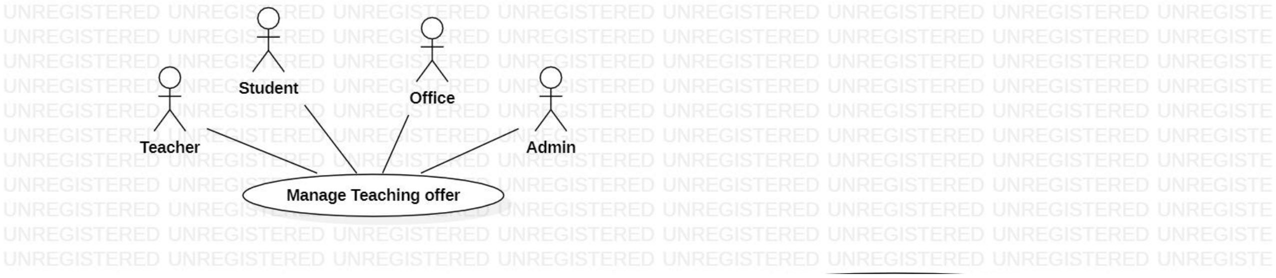
Attendance to lessons is mandatory in presence in some universities, or for some courses.
The teacher, or someone in her behalf, has to check the attendance and record present / absent students. This has to happen for each lesson of the course. A student can attend the final exam only if she has attended at least 70% of the lessons.

Currently, the teacher of a defined course, for a defined lesson, receives from a university office, a paper list of students supposed to attend the lesson. At a certain point during the lesson the teacher calls each student, following the list, and, upon answer of the student, records on the list the attendance. At the end of the lesson the teacher delivers the list to the office. At the end of the course the office computes the list of students who are granted to attend the final exam (>= 70% of lessons attended).

Since this procedure is both time consuming and not reliable, a client server application is developed.
Each student is assumed to have a smartphone, and, installed on the smartphone, a client app from the university. At a certain point during the lesson the teacher announces a one time code. The one time code is generated, upon request of the teacher, by the client app, teacher version. Each student enters, in a specific page of the app, the one time code, within max 1 minute from its announcement. The code is sent to the server, that in this way builds the list of students attending the lesson. This is repeated for each lesson, so at the end of the course overall attendance, and students who can attend the exam, is computed. To avoid cheating the smartphone app sends, with the one time code, the id of the wifi hotspot to which the smartphone is linked. The server checks that the id of the hotspot corresponds to the one(s) of the classroom in which the lesson happens.
At the end of the course both teacher and office can analyze the trend of attendance and define students who can attend the exam.

In the following consider the client server application for supporting class attendance checking.
(PS. In a real case, ex Polito, students already have a university app, and the server already exists. For simplicity here assume to build a separate client server application, not connected with the existing university application.)

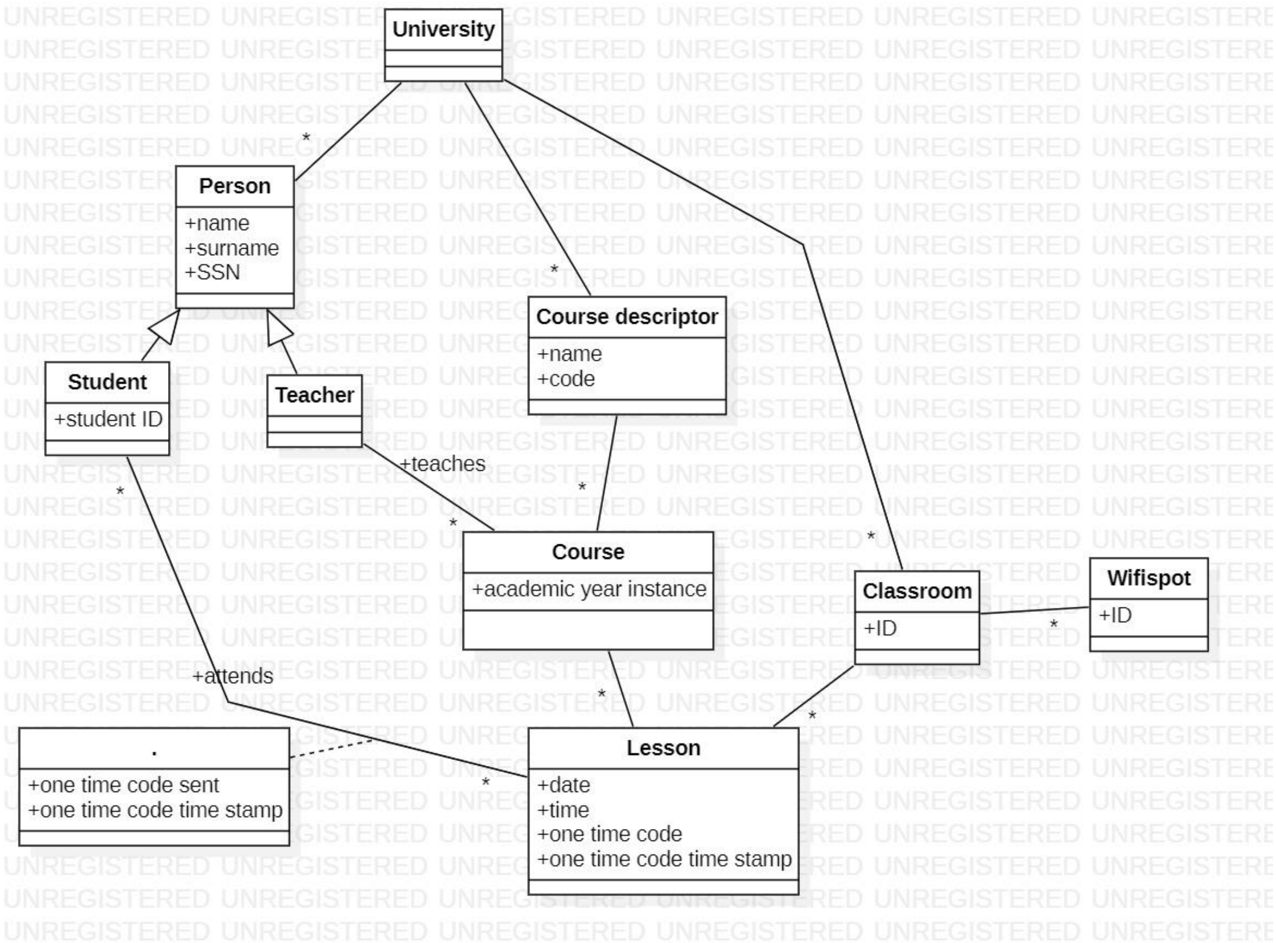
1 Define the context diagram



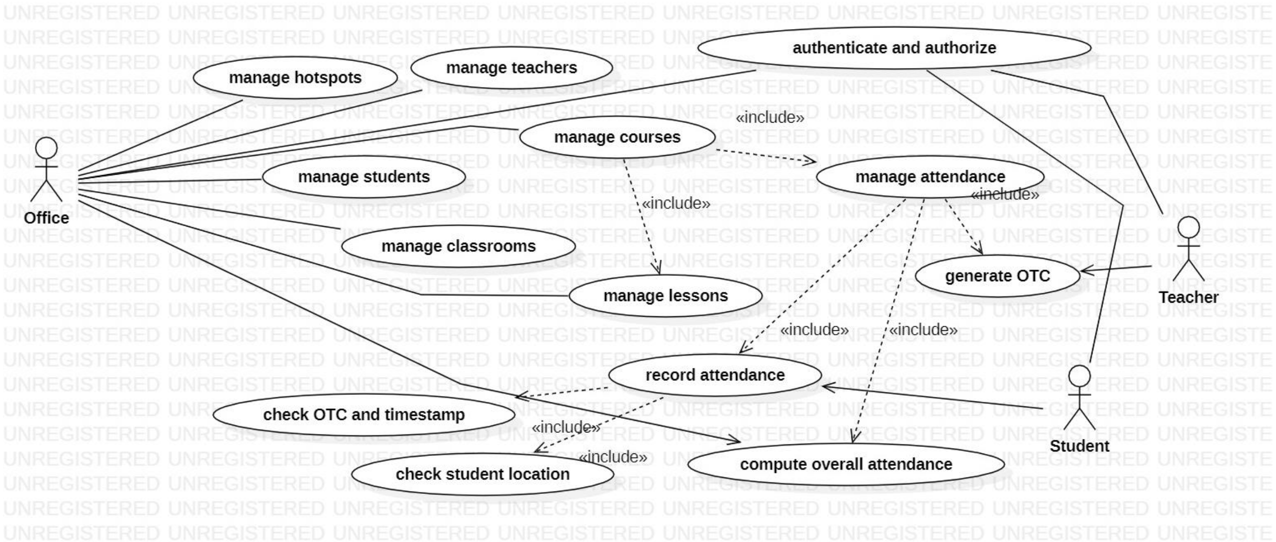
2 Define interfaces

Actor	Physical interface	Logical interface
Teacher	Smartphone	App and GUI for teacher
Student	Smartphone	App and GUI for student
Office employee	PC	Web app and GUI for employee
Admin	PC	Web app and GUI for admin

3 Define the glossary (key concepts and relationships), using a UML class diagram



4 Draw the use case diagram. For each use case give self explicable long names

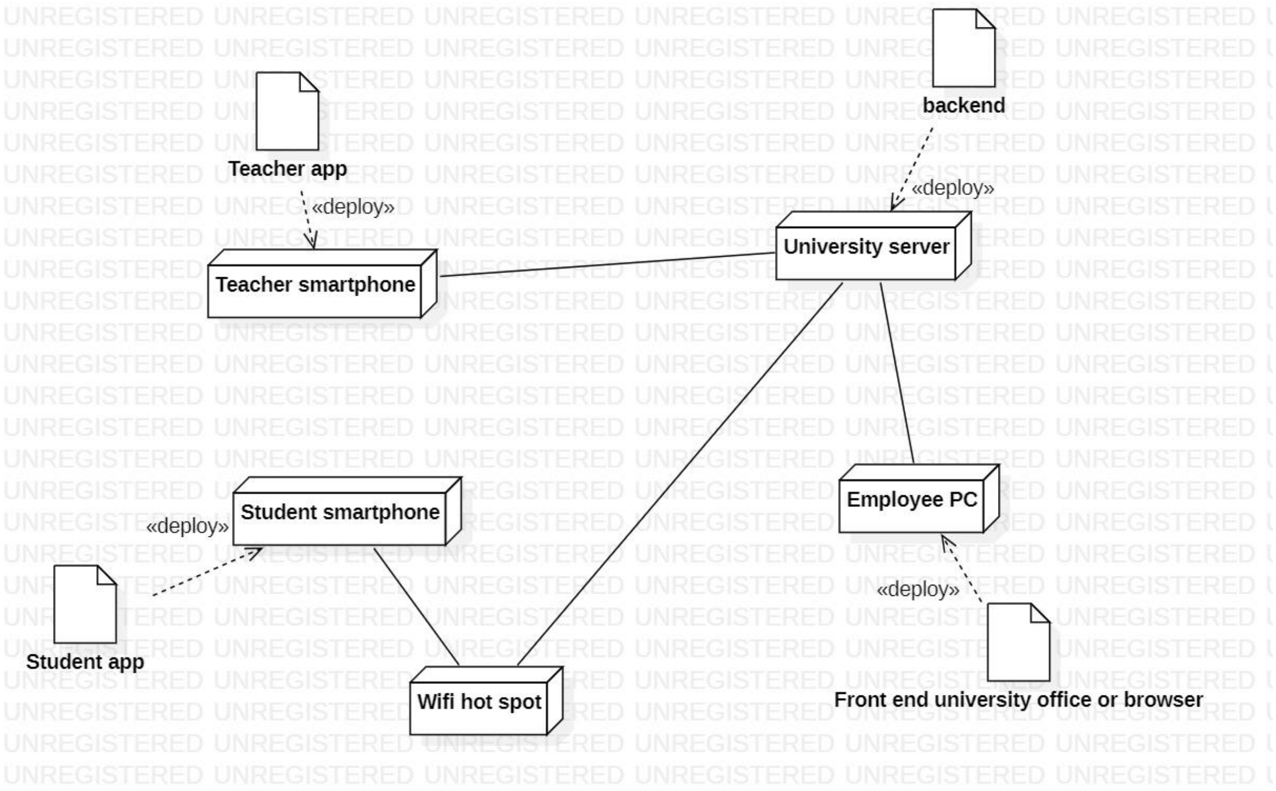


UC manage student, lesson, course, classrooms etc could be expanded (via include) with the respective CRUD student, course etc. For simplicity only UC manage attendance has been expanded.

5 Define the Non Functional requirements

Usability: students and teacher should be able to use the app with no training
Privacy: A student should only be able to access her data
Teachers and office can see data of all students
Nobody else should access any data
Efficiency: all functions should complete in <0.5 sec

6 Draw the deployment diagram



8 For the following function define test cases to obtain the highest possible node coverage, edge coverage, multiple condition coverage, loop coverage, path coverage.

For the test cases, **write only the input value**.

```
int f (int num1, int num2, int num3)
     {
     if ((num1 == num2) || (num2 == num3)) printf("no ok“);
     while (num2 < 0)
          {
          num1 = num1 - 1;
          num2 = num2 - 1;
          }
     if (num1 >= num3) printf("ok“);
     else printf("no ok“);
     }
```

Coverage type	Feasibility (Y/N)	Coverage obtained (%)	Test cases
node	Y	100	

edge	Y	100	T1(2,2,2)
			T2(1,-2,3)
			T3(1,2,3)

multiple condition (line 3)	Y	100	TT T1 TF T4(2,2,3) FT T5(3,2,2) FF T2, T3
Loop (line 4)	N	66 % (one loop not feasible)	Many T2 None T1, T3, T4 T5
path	Not feasible because of bug in while. Also fixing this bug the number of paths is high (1+ num2*2) with num2 depending on the number of bits used to represent an integer (if 32 bits, 2^31)		

A negative num2 produces an endless loop