Suppose, you are tasked to design a University Management System (UMS) for ABC University. The users of this UMS have some common attributes like, id (which will be auto generated), name, email and password. The users are either students or teachers. Students have a distinct attribute – CGPA. Teachers also have a distinct attribute – designation (Lecturer/ Assistant Professor/ Professor etc.). When a student login to his/her account, he/she only can see his/her personal account information (id, name, email, password, CGPA). But when a teacher login to his/her account, he/she can see the information of all the students (excluding password), information of all the teachers (excluding password). Not only that, a teacher can also edit the names of the students.

Your task is to design a console-based UMS implementing the above-mentioned features. First you will insert some students and teachers into the system. Then you will login as a student and check if all the features are working properly or not. Lastly you will login as a teacher and check all the features.

A demo:

```
How many students: 1
How many teachers: 1
Enter student
information:
Name: Abc
Email: abc@abc.co
Pass: 111
CGPA: 3.6
Enter teacher
information:
Name: Xyz
Email: xyz@xyz.co
Pass: 222
Designation:
Professor

Login as: 1. Student
2. Teacher
1
Email: abc@abc.co
Pass: 111
Login Successful !
Options: 1. See
Details 2. Logout 3.
Exit
1
Student Details:
ID: 1
Name: Abc
Email: abc@abc.co
Pass: 111
CGPA: 3.6
2
```

```
Login as: 1. Student
2. Teacher
2
Email: xyz@xyz.co
pass: 222
Login Successful!
Options: 1. See All
Students 2. See All
Teachers 3. Edit
Student 4. Logout 5.
Exit
3
Enter Student Id: 1
Enter New Name: Ali
Options: 1. See All
Students 2. See All
Teachers 3. Edit
Student 4. Logout 5.
Exit
1
All Student Info:
ID: 1
Name: Ali
Email: abc@abc.co
CGPA: 3.6
```

Note: You have to use nested Structure.