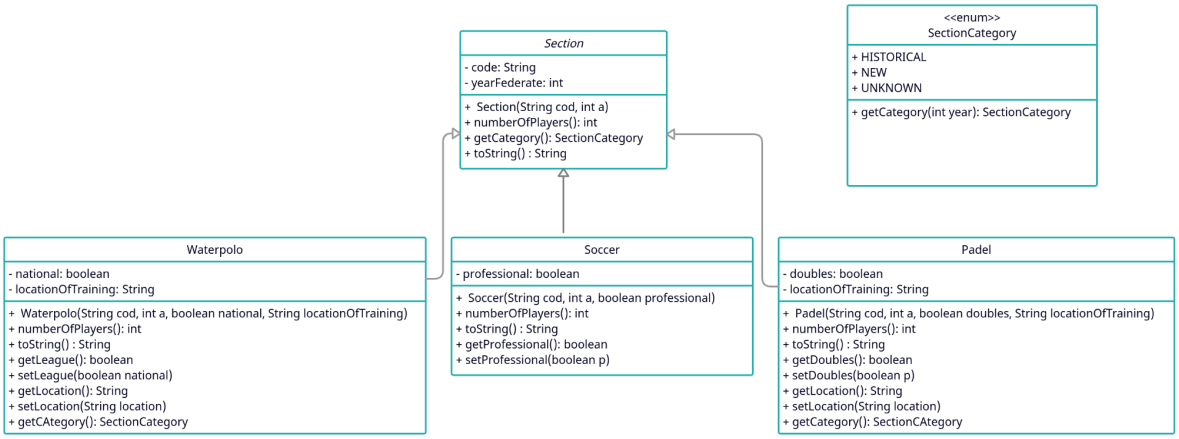
Software Analysis and Design 2021-2022
Assignment 2
Object Oriented Design
Pablo Sancho & Daniel Varela

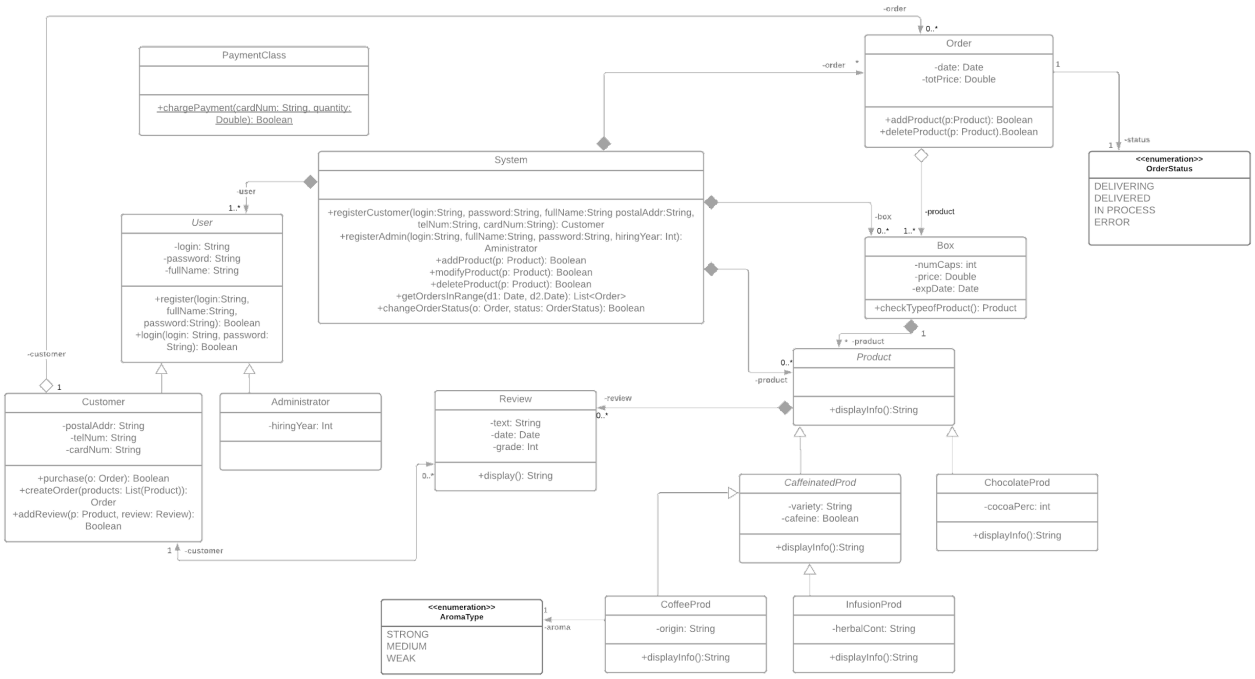
Exercise 1:



In this new class diagram we added the two new classes required (Waterpolo and Padel) and their corresponding new methods and attributes.

Exercise 2:

a,b)

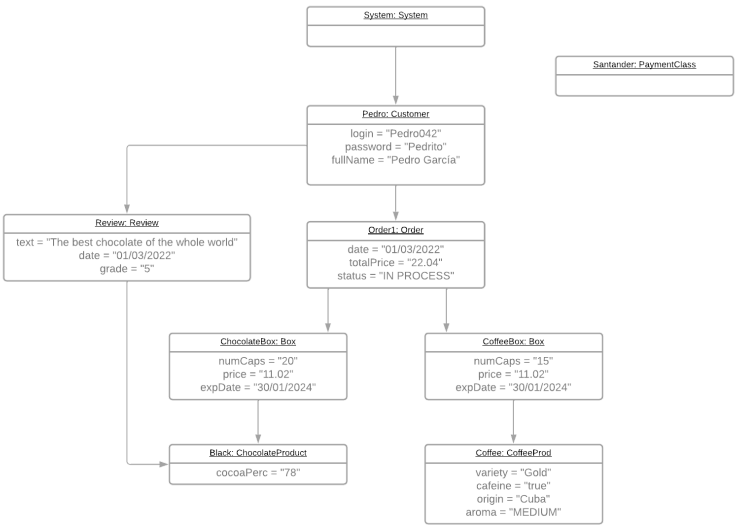


*Zoom it to see it better, also in the corresponding folder there's a png with the full diagram.

This is the class diagram of the second exercise, where we can see that it is about an e-commerce application dedicated to the sale of products, which in this case are capsules of different types.

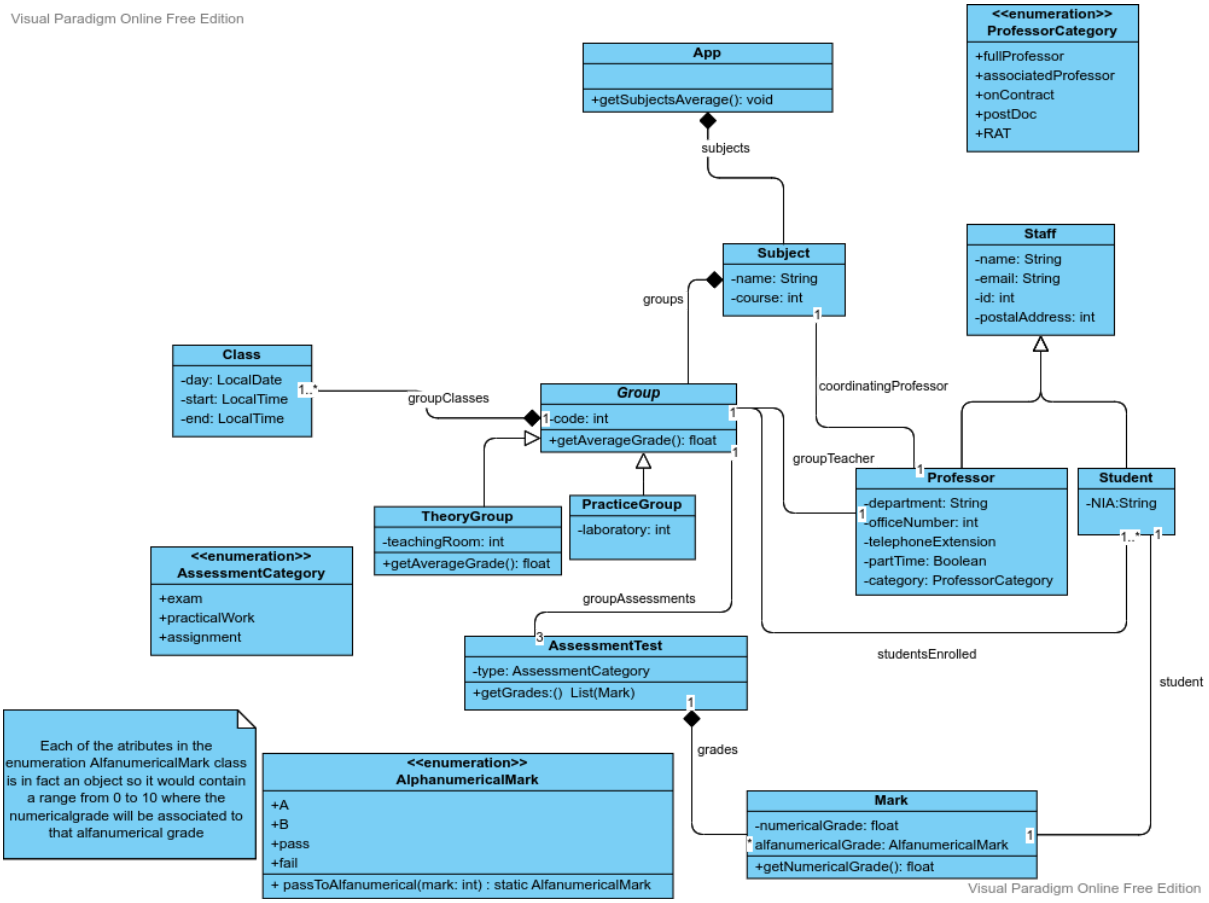
To design it we based on the instructions given to us and we added some methods and attributes to guarantee that the whole app works properly and to ensure an easy understanding.

c) This is the object diagram as stated in the exercise, and we based on our upper class diagram to design this example.



Exercise 3:

a)



Here we have the class diagram for exercise 3.a. of an application to manage the academic part of a university with all the corresponding classes to accomplish what is specified in the statement.

b)

The pseudocode of the required functions is the following:

b.1)

An override of the method *getAverageGrade* in the class TheoryGroup (so we only get the result of the theoretical assignments):

```
@Override float getAverageGrade () {
        float totalSum=0
        int totalPeople=0
        for (AssessmentTest a : this.groupAssessments){
                if (a.type == AssessmentCategory.Exam){
                        for (Mark grade : grades){
                                totalSum+=grade.numericalGrade;
                                totalPeople+=1;
                        }
                }
        }
        return totalSum/totalPeople; }
```

b.2)
A method to get the different subjects name and average grade by using the function requested in section 3.b.1.

```
void getSubjectsAverage(){
	String output="";
	float sGrade=0;
	float theoryGroupsInSubject=0;
	for (Subject s i: subjects){
		sGrade=0;
		theoryGroupsInSubject=0;
		output+=;
		for (Group g : groups){
			sGrade+=g.getAverageGrade();
			theoryGroupsInSubject+=1;
		}
		output+=s.name+" "+sGrade/theoryGroupsInSubject+"\n";
	}
	system.out.printn(output);
}
```