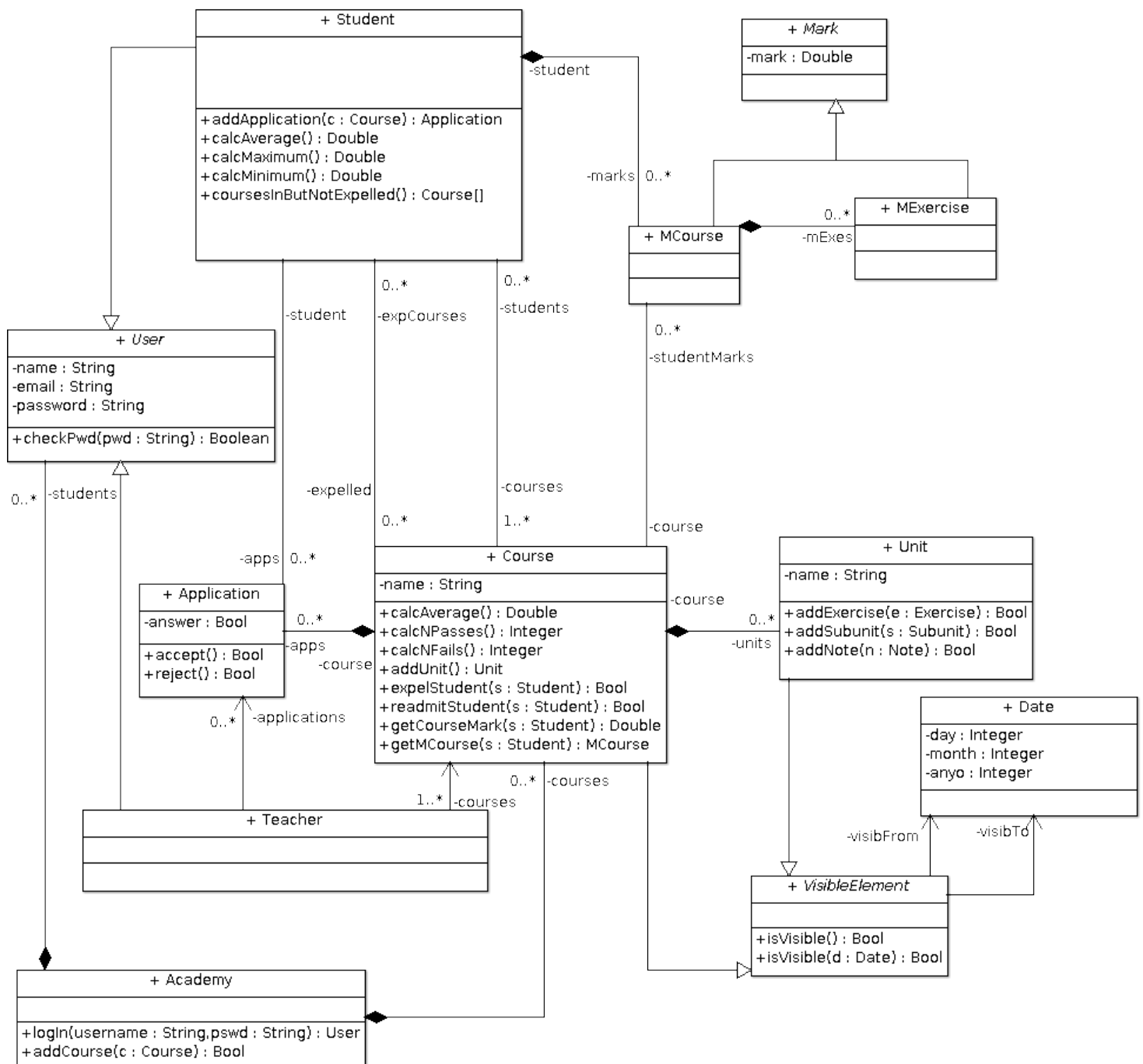


# Desing document

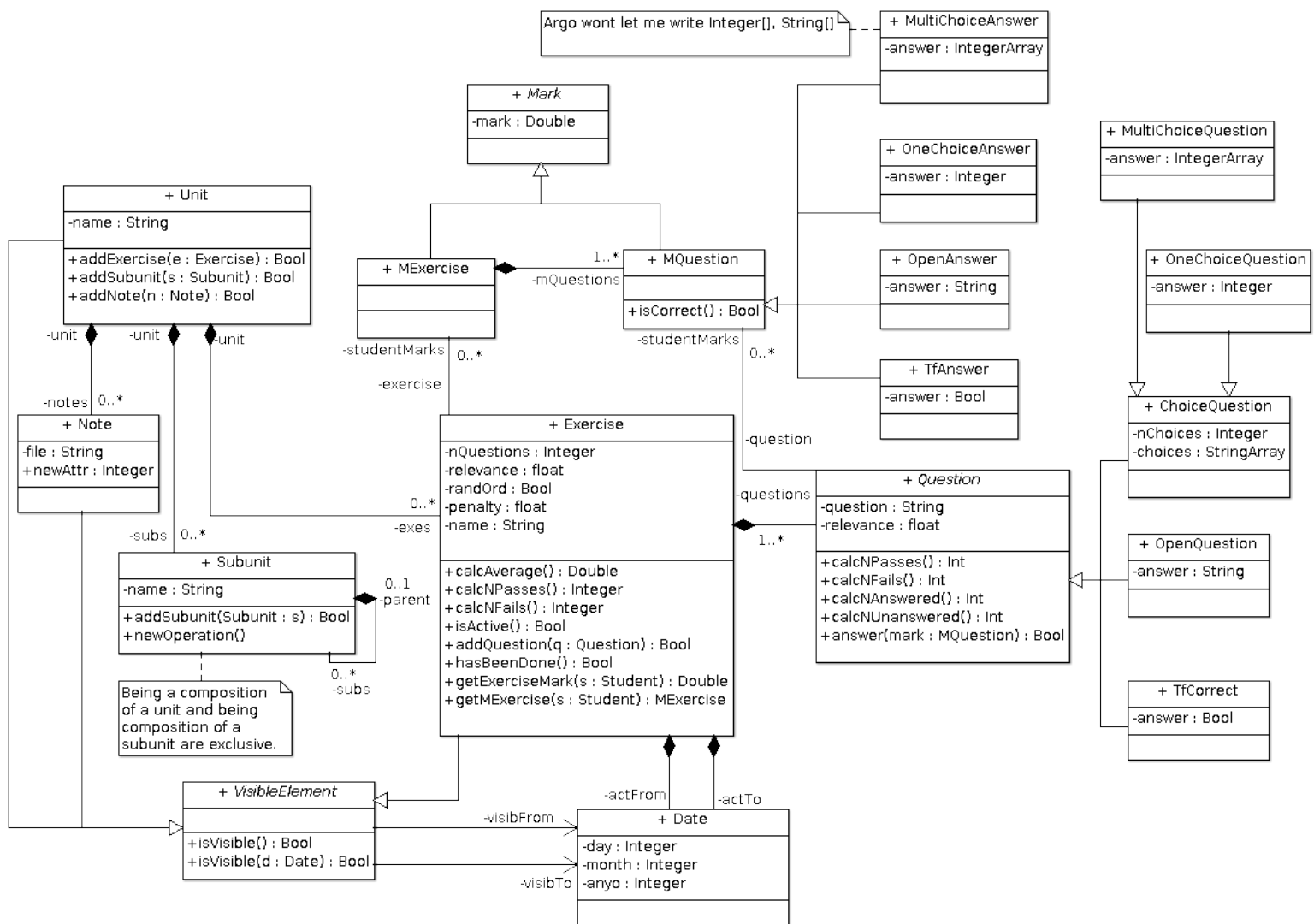
*Padsof*  
*Juan Riera Gomez*  
*Lucia Asencio Matrn*

## Class Diagram:

### Part 1



## Part 2

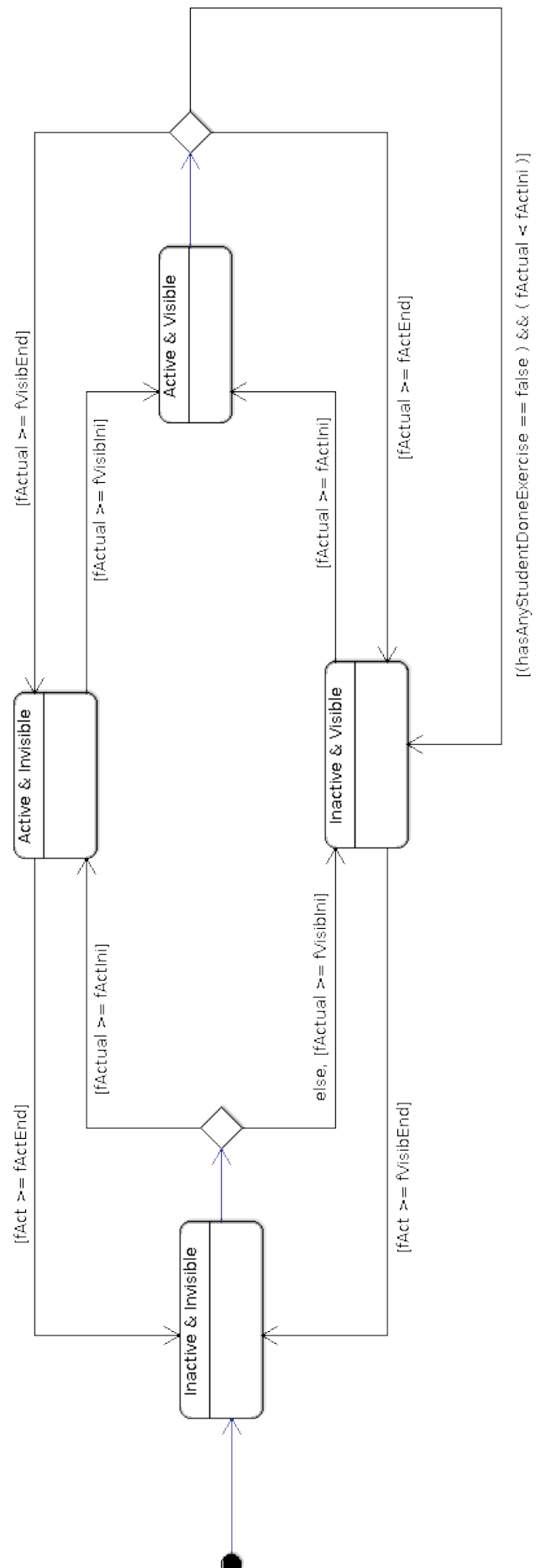


We had to divide the diagram because of its size, which would not fit in this page. Due to this, there are some repeated Classes between both parts.

## State transition diagram 1:

### Course

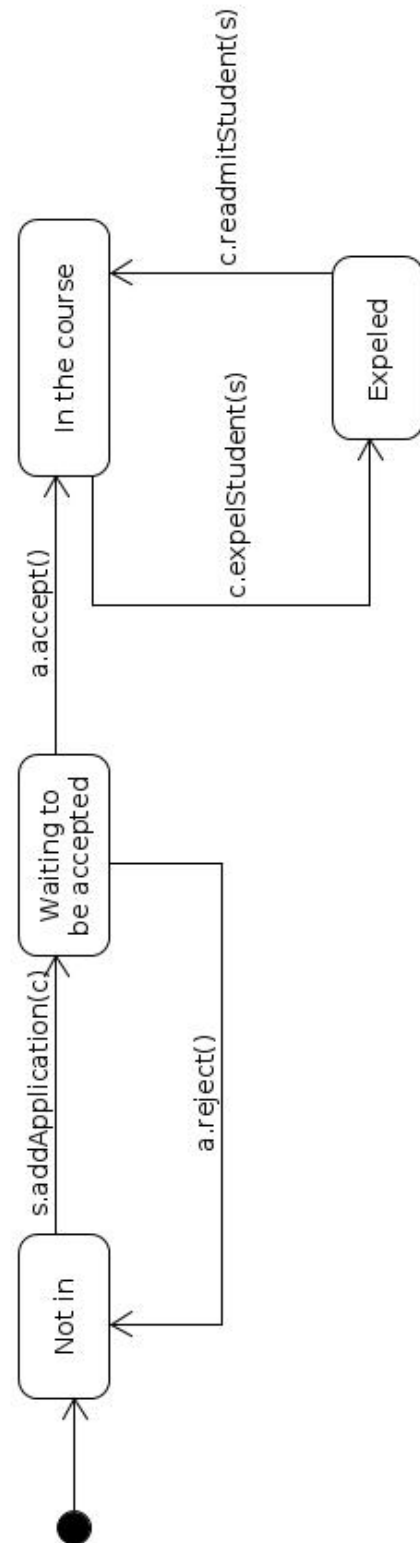
We restricted the state transition of an exercise to the 4 transitions we thought were more sensible. Therefore, transitions where visibilityEnd was before visibilityIni or activityEnd was before activityIni have been discarded. Also, we discarded the ones where the exercise was never doable (i.e., active and visible): for example, the ones where the exercise went from active to inactive and, afterwards, from visible to invisible.



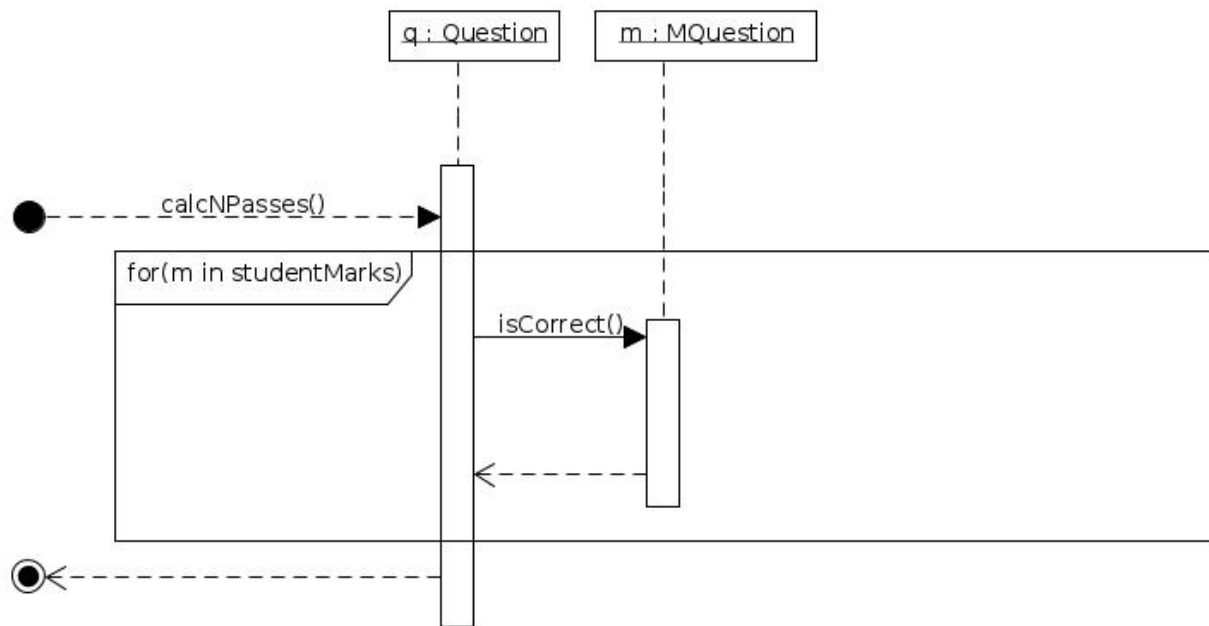
**State transition diagram 2:**  
**Possible relations between**  
**a Course and a student**

In this diagram:

- a is an application
- c is a course
- s is a student

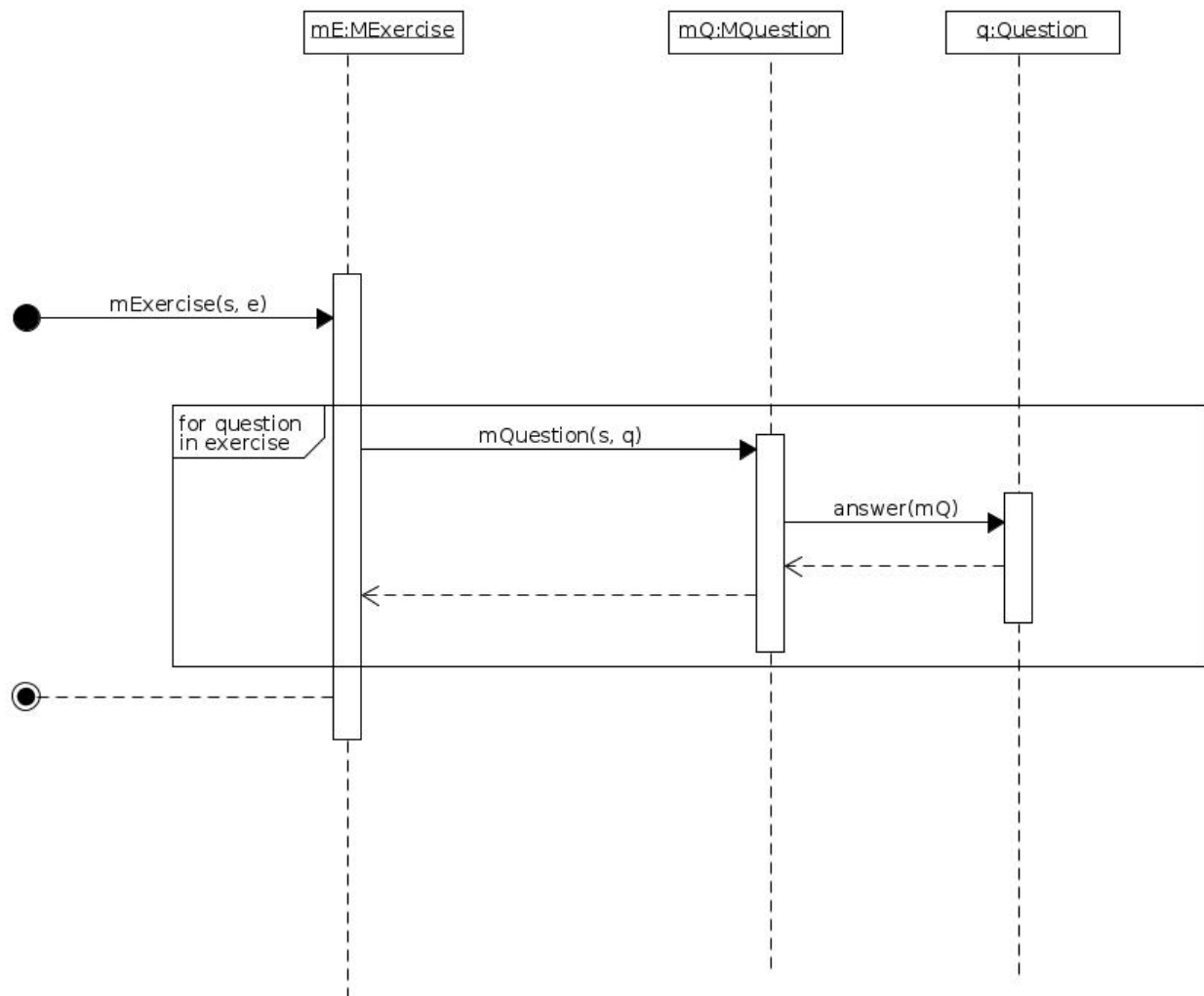


## Sequence diagram 1: Calculation of NPasses



The method `calcNPasses`, calculates the number of students who have answered a question correctly, this means that it will count the elements in `studentMarks` for which `isCorrect()` returns 'TRUE'.

## Sequence diagram 2: student solving an exercise



Sequence diagram show how a student solves an exercise. When he presses the “do exercise” button, a new “MExercise” object (instance of the exercise mark class where the student exercise, a new MQuestion object (instance of the question mark class, where the answers and mark for each question will be saved) is constructed.

## Traceability matrix associated with the teacher's requirements

We had too many methods to put them all in one matrix, so we divided it into 8 different tables.

Requirements\Methods	Stuent.calcAverage	Student.calcMaximum	Student.calcMinimum	Student.coursesInButNotExpelled	User.checkPwd
Log in					X
Accept Students					
Decline Students					
Create course					
Create unit					
Create subunit					
Create exercise					
Create question					
Create note					
Edit visibility of course					
Access student information	X	X	X	X	
Access courses list					
Access to subject statistics					
Access to exercise statistics					
Access to question statistics					
Expel a student from a course					
Readmit a student in a course					

Requirements\Methods	Academy.getCourses	Academy.login	Application.accept	Application.reject	new Course
Log in		X			
Accept Students			X		
Decline Students				X	
Create course					X
Create unit					
Create subunit					
Create exercise					
Create question					
Create note					
Edit visibility of course					
Access student information					
Access courses list	X				
Access to subject statistics					
Access to exercise statistics					
Access to question statistics					
Expel a student from a course					
Readmit a student in a course					

Requirements\Methods	course.calcAverage	Course.calcNPasses	Course.calcNFails	course.addUnit	course.expelStudent
Log in					
Accept Students					
Decline Students					
Create course					
Create unit				X	
Create subunit					
Create exercise					
Create question					
Create note					
Edit visibility of course					
Access student information					
Access courses list					
Access to course statistics	X	X	X		
Access to exercise statistics					
Access to question statistics					
Expel a student from a course					X
Readmit a student in a course					

Requirements\Methods	course.readmitStude	Course.getCourseMark	Course.getMCourse	course.setVisibilityStart	course.setVisibilityEnd
Log in					
Accept Students					
Decline Students					
Create course					
Create unit					
Create subunit					
Create exercise					
Create question					
Create note					
Edit visibility of course				X	X
Access student information		X	X		
Access courses list					
Access to subject statistics					
Access to exercise statistics					
Access to question statistics					
Expel a student from a course					
Readmit a student in a course	X				

Requirements\Methods	new Unit	Unit.addExercise	Unit.addSubunit	new Subunit	Subunit.addSubunit
Log in					
Accept Students					
Decline Students					
Create course					
Create unit	X				
Create subunit			X	X	X
Create exercise		X			
Create question					
Create note					
Edit visibility of course					
Access student information					
Access courses list					
Access to subject statistics					
Access to exercise statistics					
Access to question statistics					
Expel a student from a course					
Readmit a student in a course					



Requirements\Methods	Unit.addNote	new Note	new Exercise	Exercise.calcAverage	Exercise.calcNPasses
Log in					
Accept Students					
Decline Students					
Create course					
Create unit					
Create subunit					
Create exercise			X		
Create question					
Create note	X	X			
Edit visibility of course					
Access student information					
Access courses list					
Access to subject statistics					
Access to exercise statistics				X	X
Access to question statistics					
Expel a student from a course					
Readmit a student in a course					

Requirements\Methods	Exercise.calcNFails	Exercise.addQuestion	Exercise.getExerciseMark	Exercise.getMEExercise	new Question
Log in					
Accept Students					
Decline Students					
Create course					
Create unit					
Create subunit					
Create exercise					
Create question		X			X
Create note					
Edit visibility of course					
Access student information			X	X	
Access courses list					
Access to subject statistics					
Access to exercise statistics	X				
Access to question statistics					
Expel a student from a course					
Readmit a student in a course					

Requirements\Methods	Question.calcNPasses	Question.calcNFails	Question.calcNAnswered
Log in			
Accept Students			
Decline Students			
Create course			
Create unit			
Create subunit			
Create exercise			
Create question			
Create note			
Edit visibility of course			
Access student information			
Access courses list			
Access to subject statistics			
Access to exercise statistics			
Access to question statistics	X	X	X
Expel a student from a course			
Readmit a student in a course			

## Traceability matrix associated with the student's requirements

As it had 19 different methods, we divided the matrix in 5 with 4 methods each, so that it was understandable.

	User.checkPwd	Academy.login	Academy.getCourses	Student.addApplication
Usr/pswd log in	x	x		
Apply for a course				x
Access list of all courses		x	x	
Access his visible courses				
Access visible unit in a course				
Access visible subunits in a unit/subunit				
Access visible notes				
See visible exercises				
Solve visible&active exercises				
View exercise marks				
View course mark				
View correct vs. Student answers to questions				
View student statistics				

	Student.calcAverage	Student.calcMaximum	Student.calcMinimum	Course.getUnits()
Usr/pswd log in				
Apply for a course				
Access list of all courses				
Access his visible courses				
Access visible unit in a course				x
Access visible subunits in a unit/subunit				
Access visible notes				
See visible exercises				
Solve visible&active exercises				
View exercise marks				
View course mark				
View correct vs. Student answers to questions				
View student statistics	x	x	x	

	Mcourse.getMark	Unit.getNotes	Unit.getExercises	Unit.getSubunits
Usr/pswd log in				
Apply for a course				
Access list of all courses				
Access his visible courses				
Access visible unit in a course				
Access visible subunits in a unit/subunit				x
Access visible notes		x		
See visible exercises			x	
Solve visible&active exercises				
View exercise marks				
View course mark	x			
View correct vs. Student answers to questions				
View student statistics				

	Subunit.getSubunits()	Mexercise.getMark	New Mexercise	Question.answer
Usr/pswd log in				
Apply for a course				
Access list of all courses				
Access his visible courses				
Access visible unit in a course				
Access visible subunits in a unit/subunit	x			
Access visible notes				
See visible exercises				
Solve visible&active exercises			x	x
View exercise marks		x		
View course mark				
View correct vs. Student answers to questions				
View student statistics				

	Question.getAnswer	New Mquestion	Mquestion.getAnswer
Usr/pswd log in			
Apply for a course			
Access list of all courses			
Access his visible courses			
Access visible unit in a course			
Access visible subunits in a unit/subunit			
Access visible notes			
See visible exercises			
Solve visible&active exercises		x	
View exercise marks			
View course mark			
View correct vs. Student answers to questions	x		x
View student statistics			