Classic Testing and Quality Assurance

by Team Classic

1 Introduction

In this document we will give an overview of the measures taken to ensure the quality of our project. The first part will discuss the testing strategy and guidelines and give an overview of all the testing that was done. The second part will handle how we assured the quality of our project. We will discuss our original point of view on testing and quality assurance and how this evolved during the project.

2 Testing

In the first weeks of this project a testing strategy and test plan were defined. This test strategy turned out to be a good original definition and only a few changes were made:

- We didn't succeed in running system tests automatically so we decided these would be run by the test expert every Sunday.
- We wanted to have a code coverage as high as 80%. This turned out to be quite difficult for our project because we have code we weren't able to test such as SQL exceptions in the database that will never be thrown. This is why the test coverage is the lowest in our persistence layer. We decided our new goal was 75%. Test results can be found in 2.3.
- We decided not to implement unit tests for the frontend. Frontend is only tested with the system tests in Selenium.

2.1 Test strategy and guidelines

This section describes our final test strategy and the guidelines that need to be followed to implement this test strategy.

2.1.1 Purpose

The purpose of a test strategy is to create a shared understanding of the overall approach, tools, targets and timing of test activities.

2.1.2 Guiding principles

- Shared responsibility: everyone is responsible for testing and quality. If you write a class you should also write test code for this class.
- **Test automation**: all types of tests (unit, integration, functional, acceptance) should be automated. Manual testing will only be used for usability testing.
- **Test management**: test cases, code, documents and data will be treated with the same importance as production code. This means that test code should also be documented.
- **Test independence**: at the beginning of each test as well as at the end the database should be empty. This ensures tests will not depend on each other and a test cannot cause failures for other tests.

2.1.3 Writing test code in Java

- Each test unit must be fully independent. Each of them must be able to run alone, and also within the test suite, regardless of the order they are called. The implication of this rule is that each test must be loaded with a fresh dataset and may have to do some cleanup afterwards. This is usually handled by setUp() and tearDown() methods.
- Names of functions in test code should clearly express: the method being tested, the expected input and the expected result. Variable names should express the expected input and state. Examples:
 - public void testSum_NegativeNumberAs1stParam_ExceptionThrown(int negative_number, int positive_number)
 - public void testSum_NegativeNumberAs2ndParam_ExceptionThrown(int positive_number, int negative_number)
 - public void testSum(int positive_number1, int positive_number2)
- We will use JUnit

Example:

```
import org.junit.Test;
import static org.junit.Assert.*;

public class TestSum {
    public void testSum(int positive_number1, int positive_number2){
        int sum = positive_number1 + positive_number2;
        assertEquals(sum, sum(positive_number1, positive_number2));
    }
}
```

2.1.4 Test types

}

Test	Description	Guidelines
Unit	Testing that	Write unit test for
	verifies the	all testable classes
	implementation of	
	software elements	
	in isolation	

Test	Description	Guidelines
Integration	Testing in which	Write integration
	software elements,	tests for interaction
	hardware elements,	between controllers
	or both are	and database, and
	combined and entire interaction	
	tested until the	from API to
	entire system has	database
	been integrated	
Acceptance	Testing based on	Write Selenium
(System)	acceptance criteria	tests for all
	to enable the	acceptance tests
	customer to	
	determine whether	
	or not to accept	
	the system	

2.1.5 Test execution

Tests can be executed in Netbeans/Eclipse with JUnit. Please test the code before pushing to Git. Tests will also run automatically with Jenkins when pushed to the development branch. For some local tests a local database is needed. The section database in the readme of the projects contains instructions on how to set up a local database. Acceptance tests will not be run on the server but will be run manually every week by the test expert on Sunday. To run the acceptance tests we use the Selenium plugin for Firefox.

2.1.6 Quality and test objectives

The following quality attributes have been identified as relevant and are used as a basis for the test approach in terms of priority and test targets.

		Measure and	
Attribute	Description	Target	Priority
Usability	The platform	Real users should	Must have
	should be easy to	be asked about	
	work with	their experience	
		Adding a	
		cross-reference	
		should be easy to	
		add Asking a	
		question should	
		take at most 2	
		mouse clicks	
		Adding an	
		annotation should	
		take at most 2	
		mouse clicks	

		Measure and	
Attribute	Description	Target	Priority
Correctness	Features and	100% completion	Must have
	functions work as	of agreed features	
	intended	Severity 1 defects	
		= 0 Severity 2	
		defects = 0	
		Severity 3 defects	
		< 5 Severity 4	
		defects < 10	
Maintainability	How easy it is to	Code duplication	Must have
	add features,	< 5% Unit Test	
	correct defects or	Coverage $> 75\%$	
	release changes to	Method Length <	
	the system.	25 Lines	

2.1.7 Classifying defects

Classification	Severity	Description
1	Critical	Defect causes
		critical loss of
		business
		functionality or a
		complete loss of
		service has
		occurred
2	Major	Defect causes
		major impact to
		business
		functionality and
		there is not an
		interim
		workaround
		available
3	Minor	Defect causes
		minor impact to
		business
		functionality and
		there is an interim
		workaround
		available
4	Trivial	Defect is cosmetic
		only and usability
		is not impacted

2.2 Test list

This section gives an overview of all the tests that have been written.

2.2.1 Unit tests

Unit tests for every class in backend, every developer is responsible for the unit test code of the code that he or she writes.

2.2.2 Integration tests

Test all API methods

All these tests are in /Backend/integtestsrc/endpoint/.

Name	Description	Input	Output	Location
testGetVideo	a video can be	GET /course /1	json that	Video
	retreived	/lecture /0 /video /1	contains name, youtube url and	EndpointIT
		,	duration	
testGetVideos	all videos in a	GET /course /1	json containing a	Video
	course can be	/lecture /0	list of videos	EndpointIT
	retreived	/video		
testPostVideo	a video can be	POST /course /1	json containing	Video
	added to an existing course	/lecture /0	videoid	EndpointIT
	existing course	/video + json containing a		
		name, youtube		
		url and duration		
testUpdateVideo	an existing video	PATCH /course	-	Video
_	can be updated	/1 /lecture /0		EndpointIT
		/video $/1 + json$		
		containing a		
		name, url and		
		duration		771
testDeleteVideo	an existing video	DELETE /course	-	Video
	can be deleted	/1 /lecture /0 /video /1		EndpointIT
testVideo Not-	it is not possible	use non-existing	4xx client error	Video
FoundException	to use a	videoid		EndpointIT
	non-existing			
	videoid			
testVideo	it is not possible	use a wrong json	4xx client error	Video
BadRequest	to use a anything other than a	format		EndpointIT
	valid json format			
testGet	course notes can	GET /course /1	json that	CourseNotes
CourseNotes	be retreived	/lecture /0	contains name	EndpointIT
234100110000		/coursenotes /1	and url	
testGetAll	all coursenotes in	GET /course /1	json containing a	CourseNotes
CourseNotes	a course can be	/lecture /0	list of	EndpointIT
	retreived	/coursenotes	coursenotes	
testPost	course notes can	POST /course /1	json containing	CourseNotes
CourseNotes	be added to an	/lecture /0	coursenotesid	EndpointIT
	existing course	/coursenotes +		
		json containing a name and url		
]	name and un		

Name	Description	Input	Output	Location
testUpdate	existing	PATCH /course	-	CourseNotes
CourseNotes	coursenotes can	/1 /lecture /0		EndpointIT
	be updated	/coursenotes /1		
		+ json containing		
		a name and url		
testDelete	existing	DELETE /course	-	CourseNotes
CourseNotes	coursenotes can	/1 /lecture /0		EndpointIT
	be deleted	/coursenotes /1		77
testCourseNotes	it is not possible	use non-existing	4xx client error	CourseNotes
NotFoundExcep-	to use a	coursenotesid		EndpointIT
tion	non-existing coursenotesid			
testCourseNotes	it is not possible	use a wrong json	4xx client error	CourseNotes
BadRequest	to use a anything	format	4xx chem error	EndpointIT
Dauriequest	other than a	Tormat		Enapointri
	valid json format			
testGet	a comment can	GET /course /1	json that	Comment
VideoComment	be retreived	/lecture /0	contains	EndpointIT
, 1400 C 0111110110		/video /1	username, body,	Zinap oiii oi
		/comment /1	votes, approved,	
		, , ,	a list of replies	
			and a list of	
			selfreferences to	
			videos	
testGet	a comment can	GET /course /1	json that	Comment
CourseNotesCom-	be retreived	/lecture /0	contains	EndpointIT
ment		/coursenotes /1	username, body,	
		/comment /1	votes, approved,	
			a list of replies	
			and a list of	
			selfreferences to	
t + C - +	-11 / C	ODD / /1	coursenotes	G
testGet VideoComments	all comments of a	GET /course /1	json containing a	Comment
v ideoComments	video can be	/lecture /0	list of comments	EndpointIT
	retreived	/video /1 /comment		
testGet	all comments of a	GET /course /1	json containing a	Comment
CourseNotesCom-	coursenotes can	/lecture /0	list of comments	EndpointIT
ments	be retreived	/video /1	list of comments	Liapointii
11101105	be redictived	/comment		
testPost	a comment can	POST /course /1	json containing	Comment
VideoComment	be added to an	/lecture /0	commentid	EndpointIT
	existing video	/video /1		
		/comment + json		
		that contains		
		username, body,		
		votes, approved,		
		a list of replies		
		and a list of		
		selfreferences to		
		videos		

testPost CourseNotesComment a comment can be added to existing coursenotes testUpdate VideoComment TestUpdate VideoComment an existing comment can be updated testUpdate CourseNotesComment testUpdate VideoComment TestUpdate VideoComment TestUpdate VideoComment TestUpdate VideoComment TestUpdate VideoComment TestUpdate VideoComment TestUpdate CourseNotesComment TestUpdate VideoComment TestUpdate CourseNotesComment TestUpdate VideoComment TestUpdate CourseNotesComment TestUpdate VideoComment TestUpdate CourseNotesComment TestDelete VideoComment TendpointIT TendpointI	Name	Description	Input	Output	Location
CourseNotesComment custing comment can be updated testUpdate VideoComment testUpdate videoComment can be updated testDelete videoComment testDelete videoComment can be deleted videoComment videoCo		a comment can	POST /course /1	json containing	Comment
coursenotes	CourseNotesCom-	be added to	/lecture /0		EndpointIT
coursenotes	ment	existing	/coursenotes /1		
testUpdate VideoComment VideoCo					
testUpdate VideoComment VideoCo			that contains		
testUpdate VideoComment VideoCo			username, body,		
testUpdate VideoComment Video V					
testUpdate VideoComment Video V					
testUpdate VideoComment VideoCo					
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			selfreferences to		
VideoComment can be updated			coursenotes		
VideoComment can be updated	testUpdate	an existing	PATCH /course	-	Comment
testUpdate CourseNotesComment can be updated testDelete VideoComment video / 1 / comment / 1 + json containing username and body testDelete VideoComment testDeleteV CourseNotesComment can be deleted VideoComment / 1 / liceture / 0 / coursename and body testDeleteV CourseNotesComment testDeleteV CourseNotesComment testComment it is not possible to use a monomentid testComment testComment UnauthorizedEx-ception testComment UnauthorizedEx-ception testPostReply ToComment a rely can be deleted / comment port of the comment can be comment and manage the comment testPostReply ToComment Video / 1 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course - Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 / course Comment Video / 1 / liceture / 0 /					
testUpdate CourseNotesComment can be updated testDelete VideoComment an existing Comment can be deleted videoComment testDeleteV CourseNotesComment comment can be deleted videoComment testDeleteV CourseNotesComment comment can be deleted video lestDeleteV CourseNotesComment comment can be deleted video lestDeleteV CourseNotesComment ment testComment it is not possible to use a monon-existing commentid testComment badRequest it is not possible to use a anything other than a valid json format testComment UnauthorizedEx-ception testPostReply testPostReply are ply can be location containing username and body DELETE /course - Comment lest ouse a non ous is ting comment id video lest ouse a wrong json other than a valid json format testComment UnauthorizedEx-ception testPostReply are ply can be location containing username and body PATCH /course - Comment LendpointIT EndpointIT Axx client error Comment EndpointIT Axx client error Comment EndpointIT UnAuthorizedIT UnAuthorizedIT UnAuthorizedIT Jison containing Reply Folomment I json containing I jeson containing Reply EndpointIT					1
testUpdate CourseNotesComment ment testDelete VideoComment an existing comment can be updated testDelete VideoComment testDelete VideoComment an existing comment can be deleted body DELETE /course /1 /lecture /0 /video /1 /lecture /0 /coursenotes /1 /lectur		-F			
testUpdate CourseNotesComment ment an existing comment can be updated body PATCH /course /1 /lecture /0 /coursenotes /1 /comment /1 + /son containing username and body testDelete VideoComment deleted videoComment testDeleteV CourseNotesComment ment testComment NotFoundException testComment testComment testComment testComment non-existing commentid testComment to use a anything other than a valid json format testComment UnauthorizedException testPostReply ToComment an existing comment unauthorized user comment unauthorized user added to an unauthorized user added to an updated PATCH /course /1 /lecture /0 // video /1 /lecture /0 // video /1 /lecture /0 // coursenotes // coursenotes // coursenotes // /lecture /0 // coursenotes // // course /1 // coursen					
testUpdate CourseNotesComment ment Determinent			-		
testUpdate CourseNotesComment ment an existing comment can be updated be updated comment can be updated comment /1 /lecture /0 /coursenotes /1 /comment /1 + json containing username and body body body testDelete VideoComment comment can be deleted video /1 /lecture /0 /locmment /1 testDeleteV CourseNotesComment ment comment can be deleted /2 /lecture /0 /locmment /1 /comment /1 testCourseNotesComment ment testComment it is not possible to use a non-existing commentid testComment testComment to use a anything other than a valid json format testComment UnauthorizedEx- ception testPostReply a reply can be added to an and existing /1 /lecture /0 /coursenotes /1 /lecture /0 /video /1 /lecture /0 /1 /lecture /0 /2 /lecture /0 /2 /lecture /0 /3 /lecture /0 /4xx client error /4					
CourseNotesComment can be updated	testUpdate	an existing		_	Comment
ment updated //coursenotes /1 //comment /1 + json containing username and body testDelete an existing comment can be deleted /video /1comment /1 testDeleteV an existing comment can be deleted //courseNotesComment can be deleted //courseNotesComment can be deleted //coursenotes /1 /lecture /0 /coursenotes /1 /lecture /0 /course /1 /lecture /course /1 /					
testDelete VideoComment VideoComment testDeleteV CourseNotesComment ment testComment NotFoundException BadRequest testComment testComment testComment testComment testComment to use a non-existing comment it is not possible to use a anything other than a valid json format testComment UnauthorizedException testComment testComment testComment testComment testComment testComment testComment to use a anything other than a valid json format testComment testComment testComment testComment testComment to use a anything other than a valid json format testComment testComment testComment testComment testComment action TestComment testPostReply a reply can be replyid pocument					Enapointri
testDelete VideoComment VideoComment Comment can be deleted Video VideoComment testDeleteV CourseNotesComment ment testComment NotFoundException BadRequest to use a a valid json format testComment UnauthorizedException testComment UnauthorizedException testPostReply ToComment an existing comment can be deleted Jetter / 0 /video /locursenotes	1110110	ap acced	'		
testDelete VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment Video Vide					
testDelete VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment Video Vide					
testDelete VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment VideoComment Comment can be deleted Video V					
VideoComment can be deleted /video /lcomment /1 testDeleteV an existing comment can be deleted /lcomment /1 testComrent deleted /lcourse /1 /lecture /0 /courseNotesComment /1 testComment it is not possible to use a non-existing commentid it is not possible to use a anything other than a valid json format testComment UnauthorizedException Only the creator or an admin can manage the comment testPostReply ToComment added to an // lecture /0 /locurse /1 /lecture /0 /lecture /0 /locurse /1 /lecture /locurse /	testDelete	an existing		_	Comment
deleted /video /1comment /1 testDeleteV an existing comment can be deleted /1 /lecture /0 /coursenotes /1 /lecture /0 /coursenotes /1comment /1 testComment NotFoundException non-existing commentid it is not possible to use a non-existing commentid it is not possible to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment comment testPostReply ToComment an existing comment id it is not possible to use a wrong json format or an admin can manage the comment or an added to an /lecture /0 /lecture //lecture		_	· · · · · · · · · · · · · · · · · · ·		
testDeleteV an existing comment can be deleted /courseNotesComment to use a /l commentid /comment does /comment does /comment does /comment does /courseNotesComment /l /lecture /0 /coursenotes /lcomment /locomment /loco	, race comment				Zirap om or z
testDeleteV courseNotesComment can be deleted comment can be deleted comment /1 /lecture /0 /coursenotes /1comment /1 testComment it is not possible to use a non-existing commentid it is not possible to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment testPostReply ToComment added to an TestComment can be deleted /1 /lecture /0 /course or /1 /lecture /0 /coursenotes /1 /lecture /0 /coursenotes /1 /lecture /0 /coursenotes /1 /lecture /0 / EndpointIT TendpointIT			'		
CourseNotesComment can be deleted /1 /lecture /0 /coursenotes /1comment /1 testComment it is not possible to use a non-existing commentid testComment it is not possible to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment testPostReply ToComment added to an // I /lecture /0 /coursenotes /1 /lecture /0 /coursenotes /1comment // I /lecture /0 /coursenotes // I /lecture /0 /course /1 /course /1 /lecture /0 /course /1 /lecture // EndpointIT // EndpointIT	testDeleteV	an existing		_	Comment
ment deleted /coursenotes /1comment /1 testComment it is not possible to use a commentid commentid testComment it is not possible to use a non-existing commentid testComment it is not possible to use a wrong json format format format comment use a anything other than a valid json format testComment UnauthorizedException of the comment or an admin can manage the comment comment added to an comment format format to perform an action format forma					
testComment	ment				1
testComment NotFoundException to use a non-existing commentid testComment to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment testPostReply ToComment added to an reply can be a comment to use a non-existing comment comment only the creator of the comment added to an reply can be added to an reply can be a comment comment comment only the creator of the comment added to an reply can be added to an reply can be reply in the sex comment of the comment occurred to use a non added to an reply can be added to an reply can be reply in the sex client error comment occurred to use a wrong json and awx client error and the end of the comment occurred to use a wrong json and the end of the comment occurred to use a non authorized user to perform an action action occurred to perform an action added to an reply can be reply in the sex client error occurred to comment occurred to use a wrong json and the end of the comment occurred to use a non authorized user to perform an action occurred					
NotFoundException to use a non-existing commentid testComment to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment testPostReply ToComment added to an comment comment to use a non less the comment of the comment of the comment added to an comment comment of the comment o	testComment	it is not possible		4xx client error	Comment
testComment testComment BadRequest to use a anything other than a valid json format testComment UnauthorizedException testPostReply ToComment It is not possible use a wrong json format use a wrong json format to use a anything other than a valid json format use a non authorized user or an admin can manage the comment testPostReply ToComment action POST /course /1 json containing replyid EndpointIT InauthorizedIT POST /course /1 json containing replyid Reply EndpointIT		-			
testComment it is not possible use a wrong json 4xx client error Comment BadRequest to use a anything other than a valid json format only the creator UnauthorizedException or an admin can manage the comment comment testPostReply a reply can be ToComment added to an //ecture /0 replyid Extra Comment it is not possible use a wrong json awrong json 4xx client error and 4xx clienterror UnAuthorizedIT Language Arx client error EndpointIT Language Arx client error EndpointIT Language Arx clienterror UnAuthorizedIT Language Arx clienterror UnAuthorizedIT Language Arx clienterror UnAuthorizedIT Language Arx clienterror EndpointIT Language Arx client error EndpointIT	_				1
testComment badRequest it is not possible to use a anything other than a valid json format testComment UnauthorizedException or an admin can manage the comment comment testPostReply ToComment added to an to use a wrong json format to use a mon authorized user to perform an action to perform an action group for a reply can be added to an to perform to perform an action group for a reply can be added to an to perform an action format to perform an action group for a reply can be replyid to the comment format to use a wrong json and wrong json and format to use a mon authorized user to perform an action group for a reply can be replyid to the comment format to use a wrong json and the comment authorized user to perform an action group for a reply can be replyid to the comment format to use a wrong json and the comment format to use a mon authorized user to perform an action group for a reply can be replyid to the comment format to use a wrong json and the comment format to use a wrong json and the comment format to use a wrong json and the comment format to use a wrong json and the comment format format format format and the comment format forma					
BadRequest to use a anything other than a valid json format testComment UnauthorizedEx-ception or an admin can manage the comment comment testPostReply a reply can be ToComment added to an format format format format any lecture /0 format	testComment		use a wrong json	4xx client error	Comment
testComment only the creator of the comment ception or an admin can manage the comment testPostReply a reply can be ToComment added to an other than a valid json format use a non at the value and the valid json format use a non authorized user to perform an authorized user to perform an action action performan action replyid EndpointIT		_			
testComment only the creator UnauthorizedEx- ception or an admin can manage the comment comment testPostReply a reply can be ToComment added to an visual area of the creator authorized user to perform an action action ToComment use a non authorized user to perform an action action ToComment possible and possible authorized user to perform an action action ToComment possible authorized user to perform an action proposed possible action proposed possible authorized user to perform an action proposed possible action proposed possible authorized user to perform an action proposed possible action pro	_				
testComment only the creator UnauthorizedEx- ception or an admin can manage the comment comment testPostReply a reply can be ToComment added to an visual area of the creator authorized user to perform an action action ToComment use a non authorized user to perform an action action ToComment possible and possible authorized user to perform an action action ToComment possible authorized user to perform an action proposed possible action proposed possible authorized user to perform an action proposed possible action proposed possible authorized user to perform an action proposed possible action pro		valid json format			
UnauthorizedEx- ception of the comment or an admin can manage the comment comment testPostReply a reply can be ToComment added to an /lecture /0 replyid replyid EndpointIT	testComment		use a non	4xx clienterror	UnAuthorizedIT
manage the comment action testPostReply a reply can be ToComment added to an /lecture /0 replyid EndpointIT	UnauthorizedEx-	of the comment	authorized user		
comment POST /course /1 json containing Reply testPostReply a reply can be POST /course /1 json containing Reply ToComment added to an /lecture /0 replyid EndpointIT	ception	or an admin can	to perform an		
testPostReply a reply can be POST /course /1 json containing Reply ToComment added to an /lecture /0 replyid EndpointIT		manage the	action		
ToComment added to an /lecture /0 replyid EndpointIT		comment			
ToComment added to an /lecture /0 replyid EndpointIT		a reply can be	POST /course /1	json containing	Reply
		added to an		replyid	EndpointIT
existing comment /comment /1		existing comment	/comment /1		
/children + json			/children + json		
containing a			containing a		
body			body		

Name	Description	Input	Output	Location
testPostReply ToReply	a reply can be added to an existing reply	POST /course /1 /lecture /0 /reply /1 /children + json containing a body	json containing replyid	Reply EndpointIT
testUpdateReply	an existing reply can be updated	PATCH /course /1 /lecture /0 /reply /1 + json containing a body	-	Reply EndpointIT
testDeleteReply	an existing reply can be deleted	DELETE /course /1 /lecture /0 /reply /1	-	Reply EndpointIT
testReply Not- FoundException	it is not possible to use a non-existing replyid	use non-existing replyid	4xx client error	Reply EndpointIT
testReply BadRequest	it is not possible to use a anything other than a valid json format	use a wrong json format	4xx client error	Reply EndpointIT
testGet VideoReferences	all references friom a comment to a video can be retreived	GET /course /1 /lecture /0 /comment /1 /videoref	json containing a list of references	Reference EndpointIT
testGet CourseNotesRef- erences	all references from a comment to coursenotes can be retreived	GET /course /1 /lecture /0 /comment /1 /coursenotesref	json containing a list of references	Reference EndpointIT
testPost VideoReference	a video reference can be added to an existing comment	POST /course /1 /lecture /0 /comment /1 /videoref + json that contains videoid and timestamp	json containing referenceid	Reference EndpointIT
testPost CourseNotesRef- erence	a reference can be added to an existing comment	POST /course /1 /lecture /0 /coursenotes /1 /coursenotesref + json that contains coursenotesid, and a list of locations (pagenumber, x1,x2,y1,y2)	json containing referenceid	Reference EndpointIT
testUpdate VideoReference	an existing reference can be updated	PATCH /course /1 /lecture /0 /comment /1 /videoref /1 + json containing videoid and timestamp	-	Reference EndpointIT

Name	Description	Input	Output	Location
testUpdate	an existing	PATCH /course	-	Reference
CourseNotesRef-	reference can be	/1 /lecture /0		EndpointIT
erence	updated	/comment /1		
		/coursenotesref		
		/1 + json that		
		contains		
		coursenotesid,		
		and a list of		
		locations		
		(pagenumber,		
		x1,x2,y1,y2)		
testDelete	an existing	DELETE /course	-	Reference
VideoReference	reference can be	/1 /lecture /0		EndpointIT
	deleted	/comment /1		
		/videoref /1		
testDelete	an existing	DELETE /course	-	Reference
VCourseNotes-	reference can be	/1 /lecture /0		EndpointIT
Reference	deleted	/comment /1		
		/coursenotesref		
		/1		
testReference	it is not possible	use non-existing	4xx client error	Reference
NotFoundExcep-	to use a	referenceid		EndpointIT
tion	non-existing			
	referenceid			
testReference	it is not possible	use a wrong json	4xx client error	Reference
BadRequest	to use a anything	format		EndpointIT
	other than a			
1 1 0 1 0	valid json format	CIPITE / /1		
testGetCourse	a course can be	GET /course /1	json that	Course
	retreived		contains name,	EndpointIT
			lectures, videos	
testGetCourses	11 1	CIDITI /	and coursenotes	
testGetCourses	all courses can be	GET /course	json containing a list of courses	Course
toot Doot Course	retreived	DOCT /source		EndpointIT
testPostCourse	a course can be added	POST /course + json containing a	json containing	Course
	added	"	courseid	EndpointIT
tootIIn dataC	an origina	name		Course
test Up date Course	an existing course can be	PATCH /course	-	
		1 + json		EndpointIT
	updated	containing a		
		name, url and duration		
testDeleteCourse	an existing	DELETE	_	Course
restructore Control	course can be	/course /1	_	EndpointIT
	deleted	/ COULDE / I		Enuponiti
testAuthorisation	a course can only	use a student to	4xx clienterror	Course
ocsorrannon isanion	be managed by a	create a course	TAA CHCHICHIOI	EndpointIT
	teacher or an	create a course		Liiupoiiiii
	admin			
testGetLecture	a lecture can be	GET /course /1	json that	Lecture
obs domeounc	retreived	/lecture /1	contains name,	EndpointIT
	1001017001	,1000010 / 1	videos and	Ziiapoiiivii
			coursenotes	
			Courseirones	

Name	Description	Input	Output	Location
testGetLectures	all courses in a lecture can be retreived	GET /course /1 /lecture	json containing a list of courses	Lecture EndpointIT
testPostLecture	a lecture can be added to an existing course	POST /lecture + json containing a name	json containing courseid	Lecture EndpointIT
${ m test Update Lecture}$	lecture can be updated	PATCH /course /1 /lecture /1 + json containing a name, url and duration	-	Lecture EndpointIT
testDeleteLecture	an existing lecture can be deleted	DELETE /course /1 /lecture /1	-	Lecture EndpointIT
testGetUser	a user can be retreived	GET /user /1	json that contains name, and role	User EndpointIT
testGetUsers	all users can be retreived	GET /user	json containing a list of users	User EndpointIT
testPostUser	a user can be added	POST /user + json containing a name, a password and a role	json containing userid	User EndpointIT
testUpdateUser	an existing user can be updated	PATCH /user /1 + json containing a name, a password and a role	-	User EndpointIT
testDeleteUser	an existing user can be deleted	DELETE /user /1	-	User EndpointIT
Login test	an existing user can login	POST /login + json containing username and password	-	Login EndpointIT
credentials test	athe crednatials of the logged in user can be retreived	GET /login	json containing username and role	Login EndpointIT
Logout test	an existing user can logout	POST /logout	-	Login EndpointIT
LTI test	an tool consumer can generate a secret key that can be used to use our application	json containing secret key	new secret key to form secret key pair	Lti EndpointIT

Test all controller calls

All the controller calls that are used in the endpoints will also be tested in an integration test. This is the same test as the unit test, but it uses the real database instead of a mock-up.

2.2.3 System tests

All these tests are in /systemtests/.

Name	Description	Input	Output	File
Create account	User can create	username,	you are logged in	create_account
	an account	password and	with your new	
		role	account	
Login	User can login	username and	A user has access	login
		password	to all courses on	
			the platform	
			when subscribed.	
			A user can	
			subscribe to	
TAN	TENT C: 1	T . N. 11	every course.	1
LTI	LTI Single sign	Login on Moodle	you are now	lti
	on is supported:	and launch the	logged in as a	
	a user, logged in to another	classic	Moodle user and	
	platform with	application	can use the	
	LTI SSO support		application	
	for Classic			
	implemented, has			
	access to the			
	Classic web			
	application			
	without having			
	to log in a second			
	time.			
Create course	Admins and	name	a new course	add_course
	professors can			
	create a course			
Add lecture	Admins and	name	a new lecture	add_lecture
	professors can			
	add a lecture			
Add video	Admins and	youtube url	a new video	add_video
	professors can			
4.1.1	add a video	1 1 10		11
Add course notes	Admins and	upload pdf	new course notes	add_coursenotes
	professors can			
Cubacuiba caunas	add course notes	nama ad tha	When	subscribe
Subscribe course	All users can subscribe to	name od the existing course	When successfully	subscribe
	courses available	existing course	subscribed, the	
	on the platform		course is visible	
	on the platform		on the user's	
			home page	
Add comment	All subscribed	body and	the comment is	add_comment
	users can add	timestamp	visualized on the	
	comments on a		video	
	video at a certain			
	timestamp or on			
	coursenotes with			
	a location			

Name	Description	Input	Output	File
Reply comment	Users can reply to comments on both videos and pdfs	body	the reply is visualized under the comment	add_reply
Add cross-reference	A cross-reference can be added, this is a link to course material. The linked course material can belong to the same lecture, another lecture or another course	select course material and a timestamp or pdf location	When clicking on the link yu go to the referenced course material	add_reference
hline Upvote a comment	All users can upvote or downvote all comments and replies	Upvote a comment	the vote has augmented with 1	upvote
Approve	Admins and professors can approve an answer	approve a comment	this will give a visual indication and mark the answer as correct	approve

2.3 Test results

2.3.1 Unit tests

The total coverage of the unit tests is 76%. The instructions that are not covered are mainly due to exceptions that can't be reached and we did not test functions that upload or download pdf's in unit tests.

Layer	Code coverage
API	82.4%
Controllers	79.3%
Models	100%
Persistence	71.2%

2.3.2 Integrations tests: Controllers - Database

The coverage of these integration tests is 81.3% in the controller layer. We did not test functions that upload or download pdf's in integration tests.

Layer	Code coverage
Controllers	81.3%
Models	71.7%
Persistence	48.8%

2.3.3 Integrations tests: API - Database

The coverage of these integration tests is 76.9% for all API endpoints. We did not test functions that download pdf's in integration tests.

Layer	Code coverage
API	75.9%
Controllers	71.4%
Models	74.1%
Persistence	39.2%

3 Quality assurance

Our main source of assuring quality in this project is the use of SonarQube. We made SonarQube run once a day to asses the quality of our code, and continuously used it to keep an eye on issues with the code, test coverage, documentation and code duplication. We always tried to fix important SonarQube issues as soon as possible.

Quality of the frontend was mainly tested by hands-on testing; both with Selenium and manually. We tried going through every possible scenario to ensure the behaviour of the web app is correct. This was done by various members of our team continuously over the course of the project; by the developer while developing and by another team member after a new feature was added or before a release.