



Development and Evaluation of E-assessment platform based on Multiple Choice Questions

Pedro Miguel Póvoa Correia

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Prof. António Rito Silva
Prof. Rui Prada

Examination Committee

Chairperson: Prof. David Matos
Supervisor: Prof. António Rito Silva
Member of the Committee: Prof. Daniel Gonçalves

September 2020

Acknowledgments

I have to begin by thanking Prof. António Rito Silva for his helpful guidance, enduring understanding and relentless work in helping me develop this project as well as Prof. Rui Prada for his supervision of this thesis and coordination of the IMPRESS project. I also want to thank the Pedagogical Innovation Projects of Instituto Superior Técnico for their research grant BL342/2019 and IMPRESS project funded by EU Erasmus+ Programme, grant nr. 2017-1-NL01-KA203-035259 for their patronage of this work.

As is customary and genuinely deserved, I would like to thank my parents for my upbringing and all the opportunities that they have provided me and that have made all of my work possible. Lastly but not least, to my girlfriend, my friends and colleagues that helped me concentrate and focus on my work or provided any guidance when I needed.

To each and every one of you – Thank you.

Abstract

This thesis describes the development and evaluation of an E-assessment system, used in several university courses, that focuses on multiple choice questions. The goal is to provide students with faster feedback by giving them self-assessment tools and consequently improve their motivation and engagement with the course. On the other hand, teachers shall receive detailed result information and quiz management tools that power the student usage. The design of the system is based on established feedback principles from E-assessment and Multiple Choice Question literature and inspired by existing Learning Management Systems that utilize multiple choice questions. The system's handling and usage by students is then evaluated to ascertain student performance increases and motivation improvements.

Keywords

E-assessment; Multiple Choice Questions

Resumo

Esta dissertação descreve o desenvolvimento e avaliação de um sistema de *E-assessment*, utilizado por vários cursos universitários, focado em perguntas de escolha múltipla. O objectivo do projecto é proporcionar aos alunos *feedback* imediato e ferramentas de auto-avaliação para consequentemente melhorar a sua motivação e interesse pelo curso. Por outro lado, o professor deverá receber informação detalhada dos resultados e ferramentas de administração dos *quizzes* que possibilitem a utilização por parte dos alunos. A elaboração do sistema assenta em princípios de feedback de literatura relacionados com o uso de *E-assessment* e perguntas de escolha múltipla e inspirado por sistemas de gestão de aprendizagem que utilizem perguntas de escolha múltipla. A utilização prática do sistema é posteriormente avaliada para averiguar melhorias de desempenho e motivação por parte dos alunos.

Palavras Chave

E-assessment; Perguntas de Escolha Múltipla

Contents

1	Introduction	2
1.1	Evaluation	4
1.2	Organization of the Document	4
2	Related work	5
2.1	Advantages of E-assessment	6
2.2	Multiple Choice Questions (MCQs)	7
2.2.1	MCQs Advantages	7
2.2.2	MCQs Disadvantages	8
2.3	Feedback Principles	9
3	Popular Learning Management Systems	11
3.1	Moodle	12
3.2	Google Classroom	13
3.3	ExonlineX @MOOC	14
3.4	Socrative	16
3.5	MCQ and Feedback	17
3.6	Architectural Design	18
4	Project development	19
4.1	ETL Process	20
4.2	PostgreSQL Database	21
4.3	Server Application	22
4.4	Documentation	22
4.5	User Interface	23
4.6	Functionalities	23
4.7	Testing	26
4.8	Deployment	27
4.9	Continuous Integration	27
4.10	User Community	28

4.11 Architectural Accomplishments	28
5 Result Analysis	29
5.1 Education Improvements	30
5.2 System Usage	33
5.3 Course Engagement and Assessment Improvements	34
5.4 Usability & Performance	36
5.5 Analysis Conclusion	37
6 Conclusion	38
6.1 System Limitations and Future Work	39
6.1.1 Features from the Software Engineering Project	39
6.1.2 Question Type Expansion	40
6.1.3 Fraud detection	40
6.1.4 Infrastructure	41
6.1.5 Natural Language Processing and Machine Learning	41
A Sample Code Snippets	44
B SSL Configuration Report	49
C Additional contributions	55

List of Figures

3.1 Moodle's sections with resources	12
3.2 Student view of Moodle quiz	13
3.3 Assignment creation with a Google Form associated	14
3.4 Google Form MCQ	14
3.5 Exonline multiple choice question	15
3.6 Socrative quiz results while students answer	16
3.7 Socrative quiz creation options	16
4.1 Database schema	21
4.2 Data access layers	22
4.3 Swagger	22
4.4 Teacher editing a question	23
4.5 Teacher's list of questions	24
4.6 Teacher editing as assessment	24
4.7 Teacher editing a quiz	24
4.8 Teacher's list of student answers for a quiz	25
4.9 Teacher's list of students	25
4.10 Student's view of statistics	26
4.11 Student answering quiz and results	26
4.12 Interaction between client, server and containers	27
5.1 Average results from Software Architecture exams and mini-tests	31
5.2 Final Grades Quartiles	31
5.3 Final Grades	31
5.4 Relation between usage and final grades	32
5.5 Relation between system proficiency and final grade	32
5.6 Solved quizzes per day	33
5.7 Percentage of approved students in Software Architecture	34

Listings

A.1	JPA Question Domain Entity	44
A.2	Spring Service method to create Question	45
A.3	Spring Controller method to create Question	45
A.4	Question Data Transfer Object (DTO)	46
A.5	Axios http request	46
A.6	Question frontend Model	47
A.7	Vue Component to show Question	47

Acronyms

API	Application Program Interface
CBM	Confidence-Based Marking
CMS	Content Management System
CSS	Cascading Style Sheets
CSV	Comma-separated values
ETL	Extract, Transform, Load
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IST	Instituto Superior Técnico
JPA	Java Persistence API
LMS	Learning Management System
MCQ	Multiple Choice Question
MOOC	Massive Open Online Course
MVVM	Model–view–viewmodel
NLP	Natural Language Processing
SQL	Structured Query Language
SSL	Secure Sockets Layer
SUS	System Usability Scale
UI	User Interface
XML	Extensible Markup Language

1

Introduction

Contents

1.1	Evaluation	4
1.2	Organization of the Document	4

The increased availability of computer networks and computer literacy combined with the growing number of students in higher levels of education and reduced resources of universities has led to the growth of E-learning and E-assessment platforms as a way to efficiently tackle some of the issues in our current educational systems. The use of these platforms optimizes teachers time and schools resources, making their adoption an easy way to improve efficiency. The use of Multiple Choice Questions (MCQs) in particular has become the primary type of E-assessment question since it synergises well with the use of automation which provides faster grading, result gathering and feedback delivery to students and teachers. These advantages make the transition from paper-pen assessment to E-assessment alternatives very appealing and desired by teachers as a way to optimize resources. However, this transition needs to take into consideration a set of established feedback principles as a basis to ensure that this form of assessment is suitable to every student's learning process.

Currently, at Instituto Superior Técnico (IST) university, courses rarely provide students with self-assessment tools, merely providing previous exams to students in PDF files with the correct answer easily seen which does not correctly allow for good self-assessment. Popular Learning Management Systems (LMSs) that help students in this regard are sometimes cumbersome to setup and use due to complex interfaces which discourage teachers from using them.

The proposed system aims to provide students with a self-assessment tool based on MCQs quizzes that they can interact remotely with, at any time from any place, to evaluate their understanding of the course's curriculum. Students should obtain immediate feedback once a quiz is finished and be able to request new quizzes based on topics they wish to study. The teacher should be able to create and edit questions, attribute multiple topics to them and group them into quizzes which should have well defined start and end dates. Students results should be easy to gather once the quiz is concluded. The system has to authenticate users using the university authentication system and be generic enough so that other courses of the university can use the system.

This thesis describes the development of a new E-assessment system that applies established principles on the use of MCQ quizzes and assessment. The described system was adopted in the context of two courses of IST during two separate semesters and evaluated based on data gathered during that year from actual system usage, final student grades and questionnaires made to students. The system called Quizzes Tutor was launched in September 2019 for students of the Software Architecture course of the Information and Software Engineering Master's Degree as a completely optional self-assessment tool. During the second semester it was mainly used in the Software Engineering course of the Information and Software Engineering Bachelor's Degree, where it would continue to be used for self-assessment but additionally there were in-class quizzes that students could opt in to be a component in their final grade.

This project relates with the work of [1] that analysed the use of quizzes in the context of Software

Engineering and how they can be used "as self-regulated learning (...) to improve student's performance, engagement, motivation, attitude, and feedback, and as a tool for self-assessment and diagnostic".

1.1 Evaluation

To evaluate the success of this project during the two semesters it is fundamental to define criteria. The four criteria we will use to evaluate the system and its usage in the context of the courses are the following:

1. Did the system improve student's knowledge?
2. Did students use the system and prefer it?
3. Did course engagement and assessment improve?
4. Was the system, usable and performant enough to meet its needs?

1.2 Organization of the Document

This document is organized as follows: Chapter 1 describes the intentions of the project. In Chapter 2 related literature is described as basis for the system, focusing on analysis of E-assessment and MCQs as a method of E-assessment, including its advantages and drawbacks, followed by a series of feedback principles that should be taken into account when developing a tutoring systems. In Chapter 3 a selection of popular LMSs is analysed. Chapter 4 describes the development of the system and reasoning behind architectural choices. Chapter 5 gathers all the data from system usage, student final grades and user questionnaires to assess if the project objectives were accomplished. Finally, Chapter 6 describes future work and functionalities to be implemented and lessons learned from the development and implementation of the system.

2

Related work

Contents

2.1 Advantages of E-assessment	6
2.2 Multiple Choice Questions (MCQs)	7
2.3 Feedback Principles	9

Let's begin by defining some concepts pertinent to this field such as E-learning, Learning Management Systems (LMS) and E-assessment.

E-learning can be broadly defined as the use of electronic devices and digital resources, typically on the Internet, to conduct and support the natural process of learning and skill acquisition, for more definitions see [2]. Directly from E-learning emerged the concept of LMS [3] which is the term used to designate software applications designed not only to deliver E-learning courses, training programs and educational materials but also to allow educational related tasks such as administration, documentation, tracking of online training initiatives, classroom management and to help identify training and learning gaps by utilizing analytic data and reporting. These platform can distribute online content such as text, images, videos, animations, games, activities and other external references, that can be easily accessible for learners in or outside classrooms and both synchronously or asynchronously.

As with traditional learning methods, LMS may even have built-in assessment tools, in this case denominated E-assessment. This concept can be defined as the use of electronics to check the students understanding of a given topic and for teachers to follow that process [4]. These assessment tools can provide help at every stage of traditional assessment, which are:

- Diagnostic - used to assess a student's knowledge prior to the course;
- Formative - used to follow the students' progress during the course;
- Summative - used at the end of the course to assess if the student has met the minimum requirements of the course, which requires a reliable and secure identify verification;

2.1 Advantages of E-assessment

Regardless of the purpose of the assessment procedures, it is a fundamental component of the learning process traditionally used by teachers. For this procedure to be substituted by E-assessment, whether completely or in just one of the stages, it needs to bring enough advantages to the table for it to be viable. The most common advantages of E-assessment systems discussed in [4] and [5] are:

- Time and Cost Saving: With user friendly interfaces students are faster at performing tasks while feeling in control of the activity. Due to faster result collection, teachers also save time, improving efficiency and saving costs for schools;
- Accessibility and Flexibility: Students can access materials at anytime, from anywhere, having the power to choose where and when to access the content, complete tasks and follow the course. This also has the advantage of allowing students to do the tasks at their own pace, spending more time in the areas they feel less comfortable and advancing quickly in those they are well versed in;

- **Faster feedback:** The use of the internet and distributed software allows students to receive faster feedback once the tasks are completed due to automatically generated feedback that promotes self-assessment or due to communication channels built into the assessment platforms.
- **Additional content:** E-assessment has the potential to provide personalized, adaptive and interactive feedback, tailored to particular misunderstandings and with references to relevant module materials. This additional content can be used by students to better clarify their doubts;
- **Durability and Re-usability:** Once the LMS, the course and its materials are created this allows not only for teachers to re-use them each year but also for students to re-visit the resources when necessary. This is the main way schools can save staff time, optimizing the school's resources;
- **Adaptability and Continuous Improvement:** Since the resources are re-usable and systems provide valuable, easier to collect and easier to analyze insight about students, there is a greater opportunity for teachers to continuously adapt and improve their courses, contents and tasks to fit student's needs. Each year teachers can modify their contents and students see the updated materials;
- **Motivational Privacy:** Self formative assessment allows students to make mistakes in private in a non-judgmental and impersonal system which consequently leads to students feeling more motivated to continue using the system;

Due to these advantages, the use of E-assessment systems can be tempting to schools looking to save costs, improve teacher efficiency or simply to follow the latest technical innovations, however as discussed in [6] it is necessary to base the use of these tools in robust feedback principles aimed at encouraging effective learning to increase student performance. Misguided focus on the tools instead of how they are framed within the learning process can lead to failure in reaching the expected increase in student performance.

2.2 Multiple Choice Questions (MCQs)

MCQs are one of the most common form of E-assessment since they are particularly fast at providing feedback, marking and result gathering. With the help of electronic devices and computer networks they are incredibly attractive to teachers who wish to have a more efficient use of their time. Due to this popularity, there is ample literature [7] [8] [9] on the advantages and disadvantages of their use, which are hereafter succinctly presented.

2.2.1 MCQs Advantages

The most prevalent identified advantages in using this type of questions include:

- Automatic grading and result gathering which are easily automated by E-assessment systems, reducing costs in system development;
- Most immediate feedback to students since answers don't require complex analysis of text or symbolic notation. As soon as students finish answering, the MCQ system can display if the answer is correct or not and the desired answer. Afterwards, students have time to understand what their misconceptions were and if additional studying is necessary.
- Since there is no answer formulation, only answer recognition, questions are answered much faster allowing for a larger number of questions in the same time period from a wider variety of subjects;
- MCQs were created to bring objectivity and consistency to the assessment process which not only increases speed in grading but also provides a fair evaluation to all students regardless of the evaluator;

2.2.2 MCQs Disadvantages

As discussed in [5], the lack of homogeneous testing when evaluating MCQs, due to different questions, topics, and environments, commonly produces contradictory results when evaluating their efficiency. Nevertheless, it is relevant, when developing MCQs systems, to keep in mind the following identified disadvantages in an effort to minimize their negative effects:

- MCQs are viewed as promoting memorization, factual recollection and recognition of the correct answer instead of encouraging high-level cognitive processes which are used during answer formulation. This, however, depends on how tests are constructed and, if well designed, MCQs have the ability to evaluate higher cognitive levels;
- Well designed MCQs take time and skill to construct, especially when they are meant to evaluate higher cognitive levels. Questions should also clearly convey the goal, and objective in the context of the assessment;
- In some questions, particularly in math problems, students can find that their calculations are wrong if the result is not among the options. In some cases, such as in questions of primitives where deriving each answer leads to the expression in the question, students can use a process of elimination to find the correct answer by starting from the answers and trying to reach the expression in the question;
- Questions can also test unintended topics such as literacy skills, attention, strategy or willingness to take risks, leading to the approval of students that know little about the subject. The evaluation also has to account for the luck factor since MCQs allow guessing;

- The feedback provided through MCQs can be very limited consisting only of revealing the correct answers and predetermined feedback devised during test construction which lacks personalized to the specific needs of each student;
- Lastly, wrong answers expose students to misinformation that can influence subsequent thinking about the content;

2.3 Feedback Principles

Since feedback is the most important component of formative assessment, responsible for promoting learning and the dispelling of misconceptions, seven feedback principles [8] can be used when applied by MCQs systems.

1. Help clarify what good performance is (goals, criteria and standards);
 - To self-regulate their own learning, students must have a reasonable understanding of what is required in assessment tasks. One way to accomplish this is by allowing students to create their own MCQs so they understand the goals and criteria;
2. Facilitate the development of self-assessment and reflection;
 - This can be accomplished for instance by an open-book situation where students can self-assess and self-correct or by asking students to provide a confidence rating that would require additional reflection;
3. Deliver high-quality information to students about their learning;
 - Feedback is usually provided to students solely by the correct answer or by a general explanation not specific to the student. This feedback can be enhanced through links to other classroom activities and resources or seminars and tutorials which can be based on prior performance;
4. Encourage teacher and peer dialogue around learning;
 - By generating discussion among students, the power of MCQs is magnified allowing for different perspectives to be shared and common conclusions to be quickly reached.
5. Encourage positive motivational beliefs and self-esteem;
 - Motivation is neither fixed nor completely determined by the environment and students construct their own motivation based on their appraisal of the learning and assessment context.

However, teachers can influence this appraisal through targeted interventions such as providing many low-stakes feedback opportunities, by fostering learning communities, by focusing students on learning goals rather than marks and by linking formative tasks to summative assessments;

6. Provide opportunities to close the gap between current and desired performance;

- Provide information about the student's evolution and allow repeating failed questions and response checking so students learn and evolve from past mistakes without restricting access to their desired levels of performance;

7. Provide information to teachers that can be used to help shape teaching;

- MCQs can be used by the teachers before a lecture to assess areas of learning difficulty to better focus their efforts in these particular areas;

3

Popular Learning Management Systems

Contents

3.1 Moodle	12
3.2 Google Classroom	13
3.3 ExonlineX @MOOC	14
3.4 Socrative	16
3.5 MCQ and Feedback	17
3.6 Architectural Design	18

In this section a brief description of popular LMSs is made, with a special focus on analysing their use of quizzes and MCQs as well as their general ease of use and popular features.

3.1 Moodle

Moodle¹ is an open source LMS, vastly used around the world, in all levels of education, due to its numerous features [10] and ease of use. It provides virtually every management tool a teacher could need to manage a class, lectures and students as well as teaching tools to provide content and assessment to students. The interface, as seen in Figure 3.1, is a template based system which means, for each lecture, multiple sections are created where teachers can choose to add different elements such as resources, activities, chats, glossaries, etc. These are organized very similarly to classroom lectures with a clear sequence of the material throughout the course with multiple activities in each lecture. This organization allows a great support for blended learning (combination of online educational materials with traditional place-based classrooms) and for complete remote courses.

The Moodle platform can be freely deployed in school servers with basic LAMP capabilities without any cost and be used by any number of users. After deployment users can use the web application from any device at anytime through the Internet. Besides the free and no limitation allure that the Moodle platform can provide to schools, it's most useful features include:

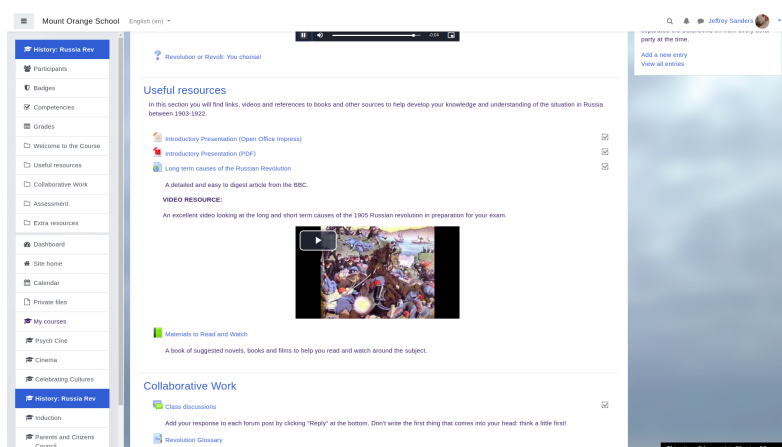


Figure 3.1: Moodle's sections with resources

- Access control to teachers and students which can be integrated with existing managements systems such as FenixEdu² as used in the course described in [11];
- Distribution of teaching content such as videos, slides, papers, websites, etc;
- Self-assessment quizzes with automatic correction with different types of questions such as fill-ins, multiple-choice, multi-choice, true-false, matching, short-answer with exact matching;
- Evaluation assignments which can then be manually corrected by the teacher;

¹<https://moodle.org/>

²<https://fenixedu.org/>

- Feedback delivery after assignments in the form of grades and forum conversations between the student and teacher after completed tasks;
- Collaborative environments (learning tasks or projects) that allows interaction between the instructor and students or among students by using different formats of social interaction.

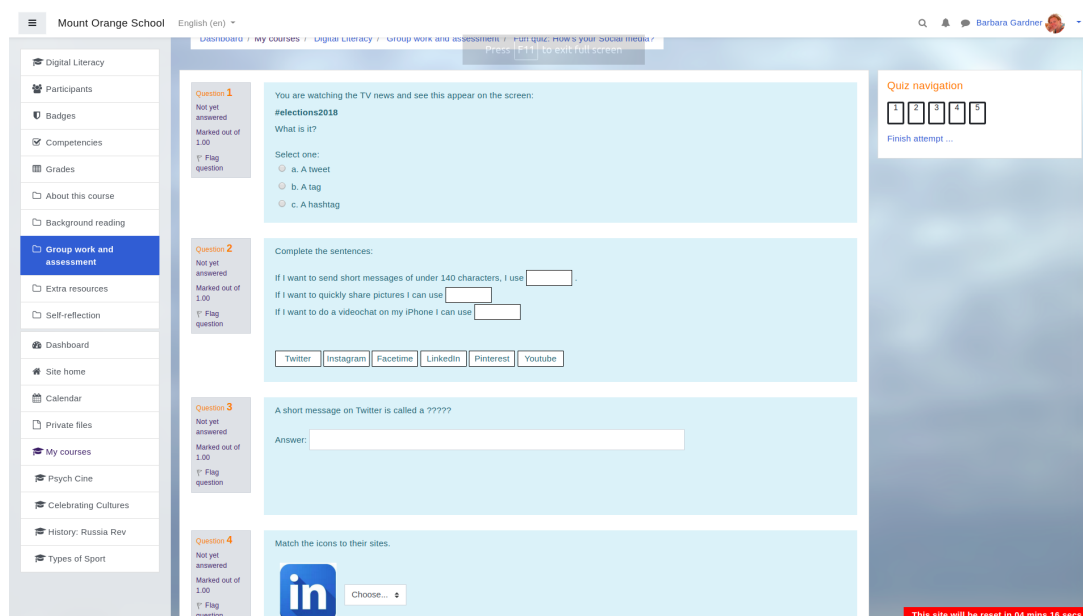


Figure 3.2: Student view of Moodle quiz

As seen in fig. 3.1, quizzes can have multiple types of questions. MCQs in particular can have multiple options such as inserting images, allowing multiple answers and adding penalty factor³

Moodle is a very complete system which makes its widely adoption, unsurprising. However, its many capabilities make its interface a bit complex for simpler use cases, specially when teachers do not require a lecture style interface and only want to focus on assessment. Another small disadvantage is requirement of local deployment which can be an issue if teachers want to quickly start using the system without limits or costs.

3.2 Google Classroom

Google Classroom is a free LMS built for the web, allowing ease of use and accessibility from all devices. It integrates with the many Google services such as Drive, Docs, Sheets, Forms, Slides and Gmail to provide a paperless and virtual classroom experience to teachers and students. It is recommended that users use Google Suite accounts, which can be cumbersome to setup but once done, can automatically import data from a school system such as FenixEdu⁴. Once students have joined, Google Classroom creates a hub that streamlines student and teacher communication.

³https://docs.moodle.org/39/en/Multiple_Choice_question_type

⁴<https://fenixedu.org/>

One of the major features of Google Classroom is the task assignment that allows teachers to easily create, distribute and grade assignments. Tasks can be defined with fixed periods that can be automatically added to the user's Google calendar. Teachers can check delivery time of submissions or lock it to not accept any more answers once the time has expired. The file sharing of these assignments is where Google Classroom really shines. Tasks can be created with multiple PDF files, Google Docs from the Google Drive, images, Youtube videos, Google Forms, etc taking advantage of the Google ecosystem.

If teachers wish to make quizzes the only way is to link a Google Forms(fig. 3.3) to the task which allows multiple types of questions, from MCQ(fig. 3.4) to input of values. Some limitations of these quizzes is that question order is forced making all quizzes equal to all students and not allowing randomization of questions or order. Using Google Forms, spreadsheets with the answers and the evaluation are automatically generated and added to the overall student grade. The down side to using Forms is that Forms is not built for teaching lacking question re-usability and integration with in class use. Tasks that involve file submissions can also be corrected on the platform itself making grading easier and after marking is done, student can in addition to the results, consult any comments that the teacher may have left. Students can then add any question to these comments fostering discussion which can be a solid form of constructive and effective feedback.

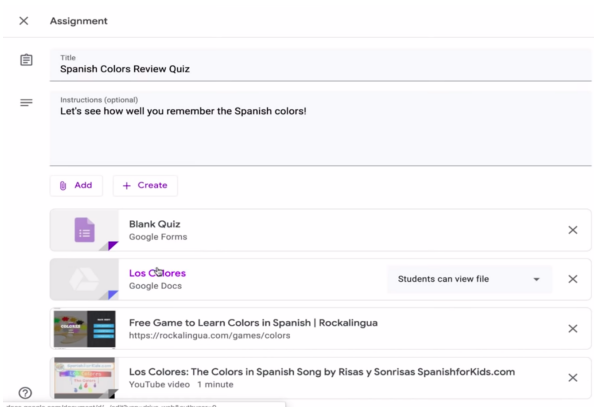


Figure 3.3: Assignment creation with a Google Form associated

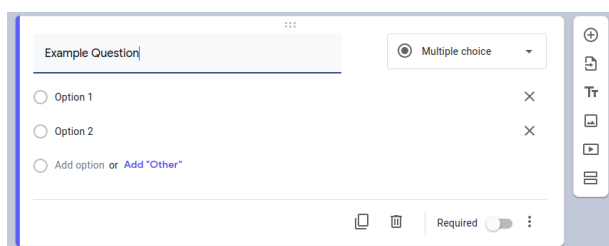


Figure 3.4: Google Form MCQ

A recently introduced feature is the Originality Report which detects sections of submitted work that contain exact or similar wording to that of other sources. This improves students writing by mentioning missing citations and provides teachers with academic integrity verification of the delivered task.

3.3 ExonlineX @MOOC

edX⁵ is one of the largest Massive Open Online Course (MOOC) providers, founded by Harvard and MIT in 2012. Its Open edX⁶ initiative released their platform as a free, open-source Content Management

⁵<https://www.edx.org/>

⁶<https://open.edx.org/>

System (CMS) allowing other organizations to deploy and host their own instance of edX to deliver their own courses rather than using edX.org. This open edX platform is currently being used in IST as MOOC Técnico⁷ since October 2016 and currently hosting 15 course. Its more recent module ExonlineX is designed for assessment and is currently being used by several courses such as Structural Mechanics with 160 students.

You are taking "Perguntas temporizadas" as a timed exam. The timer on the right shows the time remaining in the exam. To receive credit for problems, you must select "Submit" for each problem before you select "End My Exam". [Show Less](#)

End My Exam 0:04:05

< Previous

Pergunta 2

[Bookmark this page](#)

Checkboxes

1 point possible (ungraded)

You can use this template as a guide to the simple editor markdown and OLX markup to use for checkboxes problems. Edit this component to replace this template with your own assessment.

Add the question text, or prompt, here. This text is required.
You can add an optional tip or note related to the prompt like this.

☒ a correct answer

☐ an incorrect answer

☐ an incorrect answer

☒ a correct answer

Submit You have used 0 of 2 attempts

Save Reset

Figure 3.5: Exonline multiple choice question

Evaluations such as the one in fig. 3.5 can be defined with a time frame and automatically generate a spreadsheets with all evaluation results. It allows a variety of question types, from MCQ and value input to more complex responses such as mathematical expressions and chemistry formulae input. One interesting feature is that questions can generate random values for each student, preventing result sharing.

⁷<https://courses.mooc.tecnico.ulisboa.pt/>

3.4 Socrative

Socrative is a platform focused on quiz creation, free for up to 50 respondents. It has 3 types of questions: MCQ, true or false and small text input which can include images. Questions or answers can appear randomly and results can be monitored while students respond fig. 3.6, with results subsequently exported to a spreadsheet file. Authentication with FenixEdu does not exist.

State Facts

☒ Show Names ☒ Show Answers

Name	Progress	#1	#2	#3	#4	#5
Ben	100%	False	C	Idaho	False	A
Michael	80%	False	C	ID	True	
Peter	60%	True	A	TX		
Sarah	80%	True	C	Wyoming	False	
Scott	80%	True	C	Idaho	True	
Ting	40%		C	Idaho		
Class Total	60%	83%	67%	50%	100%	

Click on Question #s or Class Total %s for a detailed question view

Quizzes can be saved and later reused and there are many options when creating quizzes as seen in fig. 3.7. The “Instant Feedback” feature prevents student from rewinding to previously answered questions while the “Teacher Paced” setting allows teachers to control the rhythm in which students respond. Another gamified option is the “Space Race” where students can solve questions competitively as individuals or as a team to finish the race first. Overall the interface is very simple and intuitive and allows a very seamless experience in class allowing interaction with student in real time.

Launch Quiz

1 Choose Quiz World Facts Quiz [Change](#)

2 Choose Delivery Method and Settings Step 2 of 2

☒ Instant Feedback ⓘ
☒ Open Navigation ⓘ
☒ Teacher Paced ⓘ

☒ Require Names
☐ Shuffle Questions
☐ Shuffle Answers
☐ Show Question Feedback
☐ Show Final Score
☐ One Attempt ⓘ

[PREVIOUS](#) [START](#)

Figure 3.7: Socrative quiz creation options

3.5 MCQ and Feedback

The previous systems demonstrate the large variety of LMS that include MCQs as a form of assessment. When creating quizzes, these systems often provide a series of options for the teacher to choose from, such as number of options, possibility to return to the previous question, allowing multiple answers to the same question, presenting live results to the teacher, etc. Some of these options can considerably change the student experience but can also provide different forms of feedback. David Nicol (2007) [8] analyses four case studies and describes how their usage relates to his proposed feedback principles.

The first analysed case study is the work of Bull and Danson (2004) [12] that describes the introduction of MCQs in a course where many students were successful in coursework assignments but still failed the final examination due to delayed feedback. MCQs were introduced as a replacement for one of the assignments and it consisted of five computer-delivered multiple-choice tests related to the teaching material of the previous weeks. After each test, students received feedback on their answers enabling students to self-test their understanding (Principle 2, self-assessment) and to provide immediate feedback on their answers (Principle 3, feedback). The system also allowed the teacher to examine which questions students had performed poorly on and used this information to provide extra feedback support in those specific areas at a subsequent seminar (Principle 7, feedback shapes the teaching). Finally, the existence of multiple tests also keeps students engaged with the lectures during the duration of the course (Principle 6, motivation) and with a clearer idea of how they were progressing.

The second case study involves the use of Confidence-Based Marking (CBM) by Gardner-Medwin (2006) [13]. In CBM, students select an answer but also rate their confidence from 1 to 3, this forces them to reflect on their answer and assess their own reasoning (Principle 2, reflection/self-assessment). This score can also be used to determine the mark, so when the answer is correct the mark depends on the confidence level and if the answer is wrong a penalty can be applied so the higher the confidence level the higher the penalty. Penalties are used to compensate for the fact that students may guess the correct answers. This procedure encourages students to think deeply about their own knowledge, about whether they have a reliable justification for choosing an answer and if it is worth to risk a high penalty for high confidence. Regular use of this procedure, both in formative and in the final examinations, increases students' confidence in their knowledge and encourages regular use of these online tests (Principle 5, motivation). Importantly, CBM does not require that the teacher actually collect or analyze the reasons underlying students' answers. It is therefore surprising that it is not more widely used.

3.6 Architectural Design

The previous systems demonstrate the necessity for a easy to use web solution that can attract both teachers and students. The proposed tool should provide flexibility in its use allowing users to interact with it at any time from anywhere. Teachers should be able to easily create and edit questions and the quizzes that include them. One of the detected faults of some of these systems is the lack of focus on the re-usability of questions and quizzes. Questions should be easily used in multiple quizzes, specially when generated by students as repetitive solving ensures student proficiency in the question. Another flaw in the lack of focus on the questions is the lack of feedback to teachers regarding the questions themselves. Teachers should receive information on the difficulty of questions to better understand topics they need to readdress or questions that need to be restructured. Self-assessment options are also lacking preventing students from spontaneously generating quizzes with the specific topics they have in mind. Student results is another common feature that has to be well implement in multiple ways. Teachers should be able to have a good overlook at their class's performance while at the same time be able to see individual responses to quizzes and questions. Students, likewise, need to have feedback on their individual responses and overall performance while using the system. A good example of this feedback from the Socrative system is the in-class usage that allows teachers and/or students to see results in real time. This real time usage fosters discussion and a much faster feedback to prevent misconceptions. The developed solution like all the previous systems must provide flexibility in its use so that it can be used in multiple contexts by different systems. More specifically, the quizzes should allow self-assessment, timeless assessment and evaluation with well defined time periods. These quizzes can also have important options to assure academic integrity like question randomizing and returning to previous questions.

4

Project development

Contents

4.1 ETL Process	20
4.2 PostgreSQL Database	21
4.3 Server Application	22
4.4 Documentation	22
4.5 User Interface	23
4.6 Functionalities	23
4.7 Testing	26
4.8 Deployment	27
4.9 Continuous Integration	27
4.10 User Community	28
4.11 Architectural Accomplishments	28

At the start of the project, the existing data was comprised of 135 LaTeX files (108 mini-tests and 27 exams from the Software Architecture course from 2010 to 2017) which collectively held 1698 questions. All questions were multiple choice with four options and one correct, with some questions including an image. There was a set of excel files with the correct answers for each test and another set of excel files with the students answers to each evaluation. There was already a python script to parse and load all data from the LaTeX files into a postgresSQL database.

4.1 ETL Process

To leverage the existing data, the first step was to apply an Extract, Transform, Load (ETL) process to improve the existing parsing of LaTeX data to Structured Query Language (SQL) and to add information from the excel files with the correct option of each question and students answers. These historical records would be used to improve the model of each question regarding difficulty and to provide immediate feedback to students after answering a quiz. The parsing of LaTeX also had to be changed to convert LaTeX style commands (bold, italic, etc) to markdown. Another requirement for this ETL process was to assimilate similar questions into the same entity, as the teacher would often reuse the same question in multiple versions with different option order to prevent cheating. Sometimes the question was also reused in following years with small changes, which meant the ETL process had to be able to apply approximate string matching algorithms. To accomplish these goals, Pentaho Data Integration¹ was used and all the data was successfully loaded into an SQL database resulting in a final count of 603 unique questions from the initial 1698.

¹<https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho-platform/pentaho-data-integration.html>

4.2 PostgreSQL Database

The existing python script loaded the data into a PostgreSQL² database, a free and open-source relational database management system that fits well with the needs of this project. At the time there was a simple schema creation script that is no longer used, being the responsibility of the back-end application to create and update the existing schema. At the time of writing, the current schema looks as seen in Figure 4.1. There are different Courses that have multiple Executions, each corresponding to a different year. Questions are associated with the course so that they can be reused each year and Quizzes aggregate these Questions in the context of a Course Execution. Each Execution is also associated with multiple Users, teachers and students that can be enrolled in multiple Executions. When answering a Quiz, the response is recorded in a Quiz Answer with each individual answer to the Question recorded in a Question Answer. Questions can also be attributed a set of Topics which are then used as Topic Conjunctions when defining an Assessment. Assessments are used when students want to create a quiz to study for a specific evaluation and only want questions with a certain combination of topics to appear. During the development process there was also a need to sometimes perform data migration operations (renaming columns, changing data, etc) which were mostly done using SQL scripts.

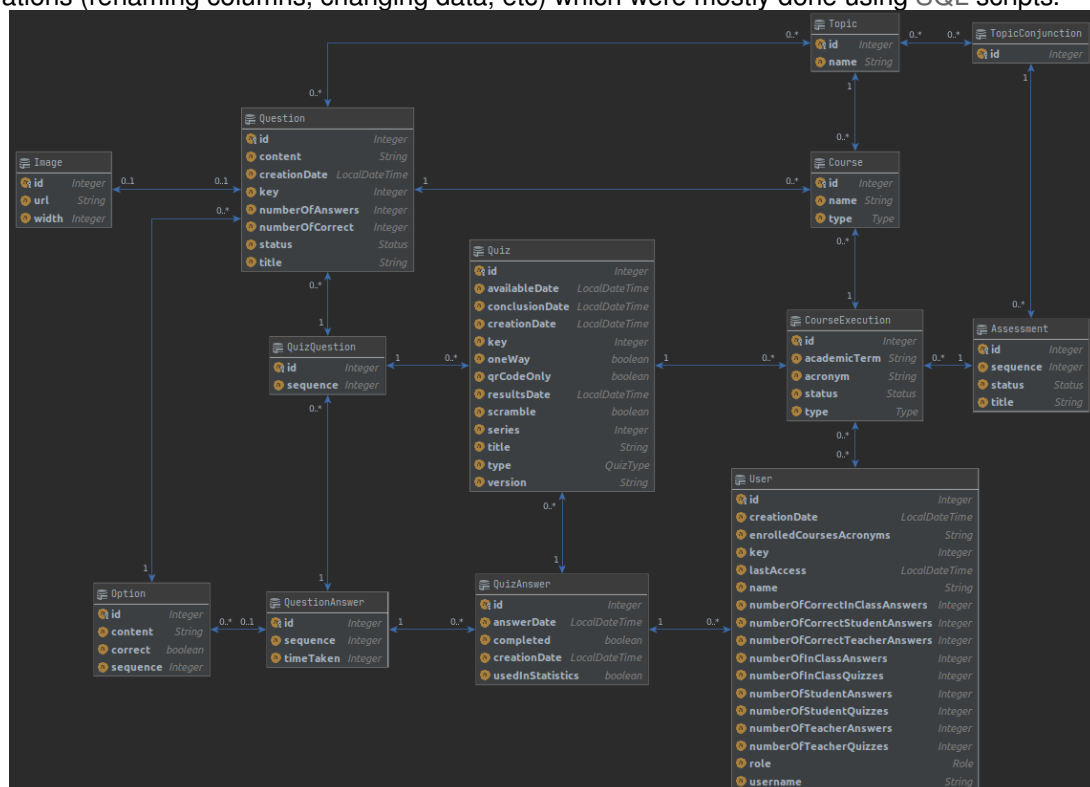


Figure 4.1: Database schema

²<https://www.postgresql.org/>

4.3 Server Application

To create the Application Program Interface (API) for the Tutor System, the Spring framework was used with Java 11. To communicate with the existing PostgreSQL database we used Hibernate's Java Persistence API (JPA), a popular Object-relational mapping for Java, which allows data from the database to be easily created and manipulated as Java objects. These objects are then used in a transactional context inside Spring services which are called by the controllers that specify the API (Figure 4.2). Sample code of these layers can be seen in Appendix A. The project is managed using Maven and to make sure that code follows strict guidelines a Java linter, SonarLint³ was used to identify and recommend changes to the code.

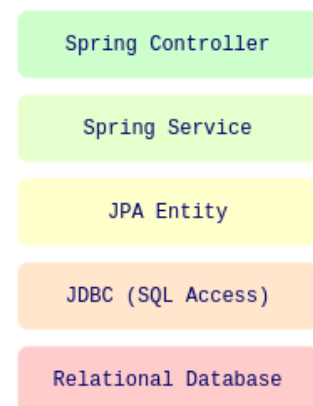


Figure 4.2: Data access layers

4.4 Documentation

To make sure the project is kept well maintained, a series of good practices was used to ensure the maintainability of the code base. These practices are important so that future developers can quickly get familiar with the existing code and quickly develop new features, without having to rewrite most of the work done so far. These practices include the use of Swagger (Figure 4.3), a library that automatically documents all existing REST API endpoints and provides an interface which also allows experimentation and initial testing. A private repository was also created with all migration scripts, deployment scripts, dumps and useful information that future administrators may need to deploy or configure the server. Finally, the Github repository README page⁴ and wiki to detail all setup requirements and useful tips that may be necessary to future developers is used, which have already been helpful for an entire class of Software Engineering students using the system as basis for the course's project.

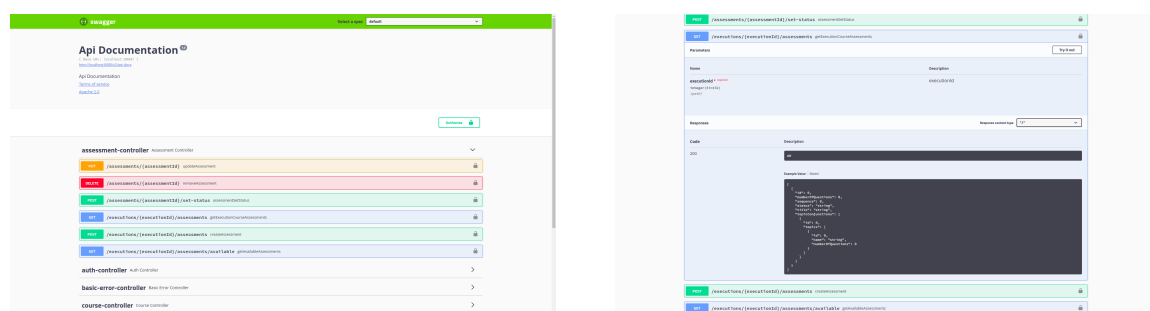


Figure 4.3: Swagger

³<https://www.sonarlint.org/>

⁴<https://github.com/socialsoftware/quizzes-tutor>

4.5 User Interface

To implement the front-end of the application we wanted a web focused framework that allowed equal use between desktop and mobile devices. We decided on Vue.js⁵, an open-source Model–view–viewmodel (MVVM) JavaScript framework for building user interfaces and single-page applications. It has a good modularity, good separation between presentation and logic, supports typescript and is built on top of Node⁶ giving developers access to the available libraries of NPM⁷. Some of the libraries currently being used include Vuetify⁸ a User Interface (UI) library full of components ready to use, Axios⁹ an HTTP client to make requests to our server and Prettier¹⁰ an automatic JavaScript linter with TypeScript and Vue support. Sample code can be seen in Appendix A.

4.6 Functionalities

The current functionalities can be seen live in <https://quizzes-tutor.tecnico.ulisboa.pt/> in a demo environment or in a video demonstration¹¹ prepared by the Software Engineering teacher António Rito Silva. For the sake of completeness a list of features with corresponding screenshots are listed below:

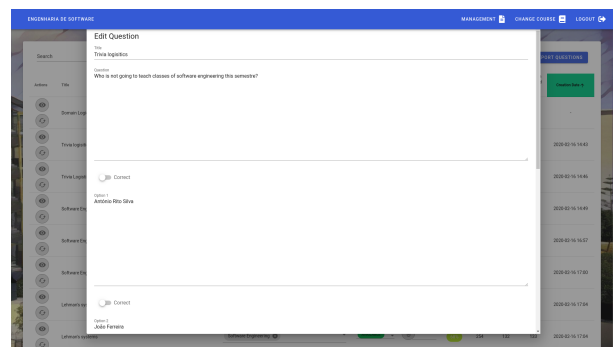


Figure 4.4: Teacher editing a question

⁵<https://vuejs.org/>

⁶<https://nodejs.org/en/>

⁷<https://www.npmjs.com/>

⁸<https://vuetifyjs.com/en/>

⁹<https://github.com/axios/axios>

¹⁰<https://prettier.io/>

¹¹<https://www.youtube.com/watch?v=PUDzoCqMznE>

Actions	Title	Topics	Status	Image	Difficulty	Answers	No. of questions	No. of times attempted	Expiration Date
	Git Commands	Software Configuration Management	Available		Easy	254	185	168	2024-02-25 16:34
	Basics State Patterns	Enterprise Application Patterns	Available		Easy	101	178	119	2024-05-01 16:30
	Software Configuration Checklist Patterns	Software Configuration Management	Available		Easy	494	180	162	2024-05-24 12:33
	Serviceable USB Pattern	Enterprise Application Patterns	Available		Easy	111	180	101	2024-04-19 16:29
	Verification and Validation	Software Testing	Available		Easy	217	147	148	2024-02-23 11:54
	Software systems	Software Engineering	Available		Easy	424	201	202	2024-02-16 11:55
	Book Case coverage	Software Testing	Available		Easy	94	170	172	2024-03-16 16:16
	Office Connectivity patterns	Enterprise Application Patterns	Available		Easy	136	171	172	2024-05-01 16:35

Figure 4.5: Teacher's list of questions

be set as available, disabled or removed, images can be uploaded and topics can be associated. Once created, all questions can then be exported to a Comma-separated values (CSV) file.

The topics can be created in a separate view and are important to classify questions and help when searching for a question, but the main purpose is to allow the creation of assessments. Assessments help students and teachers divide the curriculum in smaller portions making studying and evaluating easier. In the assessment view (Figure 4.6), teachers can create an new assessment and then instead of specifying all questions belonging to each assessment can simply use the existing topics. However, since questions can have multiple topics and an exam may have only part of them, a list of all existing combinations of topics is presented which can be added to the assessment. It is possible to see all questions of a specific combination and in the end, teachers can see the complete list of questions belonging to that assessment that will be used by students when generating quizzes for that assessment.

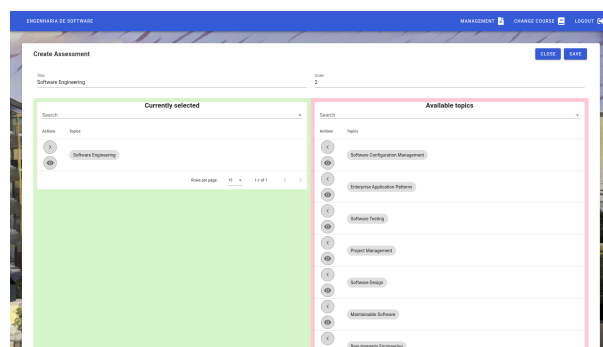


Figure 4.6: Teacher editing as assessment

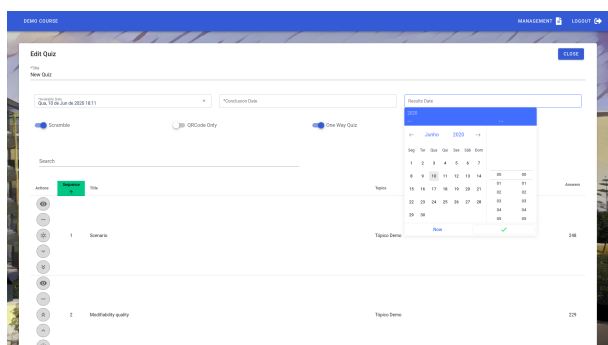


Figure 4.7: Teacher editing a quiz

so that only students with the QR code can start the quiz, can be timed making a timer appear when the

All teachers with FenixEdu credentials can login in the system where a list of their courses is presented to them, they can then activate any course so that students enrolled in that course may begin using the system. After activating the course, teachers can add new questions as seen in Figure 4.4 with full markdown support to add style, images, quotations, etc. In the question listing (Figure 4.5) each question can then

Besides randomly generated quizzes based on assessments, teachers can also create quizzes with specific questions. In the quiz edition view seen in Figure 4.7 teachers can choose the date and time when the quiz will be made available, the list of questions, and four configurations for the quiz. The quiz can be scrambled, making the order of the questions different to each student, can be QR code only, can be timed making a timer appear when the

student is solving the quiz (in this case, teachers must also specify the end date of the quiz) and can be One Way making students unable to return to the previous question when responding to the quiz.

Once one of these quizzes is answered, teachers can see a table with each student's answers and answer date (seen in Figure 4.8) which can then be exported to Extensible Markup Language (XML), LaTeX or CSV. Teachers can also see a list of the students using the system with corresponding statistics such as number of quizzes answered (generated and teacher created) and % of correct answers (generated and teacher created) which can be easily sorted using the table headers (seen in Figure 4.9).

Name	Username	Start Date	Answer Date	Answers
Student 675		2020-01-20 23:12	2020-01-20 23:12	A B A B A
Student 674		2020-01-08 18:45	2020-01-08 18:45	A D A B A
Student 672		2019-10-07 16:39	2019-10-07 16:39	A D C B B
Student 671		2019-10-07 14:15	2019-10-07 14:15	B C A B B
Student 670		2019-10-07 11:12	2019-10-07 11:12	B C D A C
Student 669		2019-10-06 20:49	2019-10-06 20:49	A C A A A
Student 668		-	-	X X X X X
Student 667		2019-10-07 14:28	2019-10-07 14:28	A D D B B
Student 666		2019-10-07 11:39	2019-10-07 11:39	A D A A A
Student 665		2019-10-06 22:54	2019-10-06 22:54	B A A D A
Student 664		-	-	X X X X X
Student 663		2019-10-07 15:13	2019-10-07 15:13	A D A B B
Student 662		2019-10-07 10:17	2019-10-07 10:17	A X D B B
Student 661		2019-10-07 10:56	2019-10-07 10:56	A C B B B
Student 660		2019-10-06 22:08	2019-10-06 22:08	A D A A A
Student 659		2019-10-07 13:16	2019-10-07 13:16	A D D B B
Student 658		-	-	X X X X X

Figure 4.8: Teacher's list of student answers for a quiz

Name	Teacher Quizzes	Generated Quizzes	Total Answers	Correct Answers	Answers Teacher Quiz	Correct Answers Teacher Quiz
Demo Student	55	463	3202	35%	206	16%
Student 653	18	0	91	52%	89	84%
Student 635	18	0	87	52%	85	84%
Student 633	18	0	92	55%	90	71%
Student 669	18	0	92	51%	90	52%
Student 648	18	0	91	51%	89	52%
Student 650	17	0	87	54%	85	55%
Student 643	18	0	92	51%	90	89%
Student 674	18	0	86	41%	84	42%
Student 661	18	0	92	50%	90	52%
Student 660	11	0	47	55%	45	57%
Student 662	18	0	72	55%	70	60%
Student 665	15	0	71	45%	69	46%
Student 638	18	0	92	55%	90	64%
Student 668	4	0	22	27%	20	30%
Student 636	0	0	2	0%	0	0%
Student 623	18	0	72	77%	70	80%

Figure 4.9: Teacher's list of students

From the student's side, there are three ways to start a quiz. There is a list of available quizzes where students can see the quizzes proposed by the teacher, a quiz generating interface where students can specify the assessment and number of questions for the quiz and a QR code view that connects to the device's webcam to read a QR code. Once a student has started a quiz as seen in Figure 4.11, they can choose the correct option and navigate between questions until they decide to complete it. After solving the quiz the results are immediately displayed if the quiz is not timed. These results can later be seen when studying in a solved quizzes list view. Lastly, students can see a stats page with the number of solved quizzes and questions, % of questions seen and % of correct answers (Figure 4.10).



Figure 4.10: Student's view of statistics

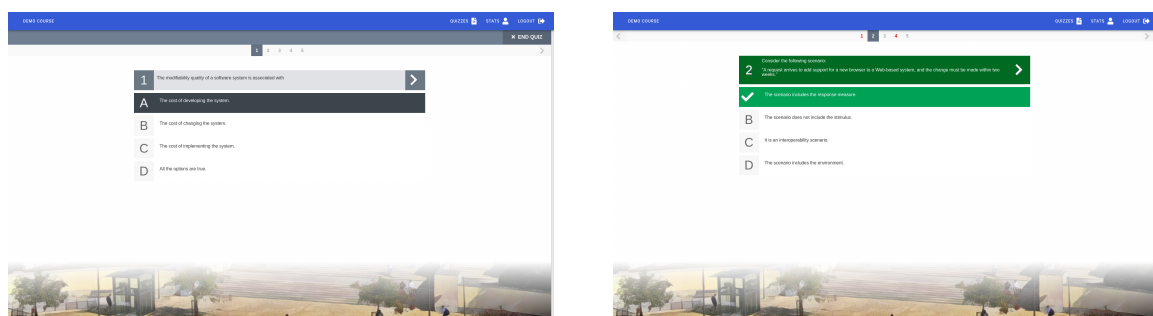


Figure 4.11: Student answering quiz and results

4.7 Testing

As a full fledged software project, it would not be complete without testing. Software tests are a integral part of any project to assure programmers and project managers that the software fulfills its requirements and that future changes do not insert any defects to the existing functionalities. To test the application services, the Spock framework is being used which provides some interesting features like Data Tables and Method Unrolling which allows a single method to run multiple test cases. Currently there are 242 tests with a 55% code coverage. Additionally there are load tests using Jmeter which test the performance of the server and finally there are end-to-end Cypress tests that test the back-end logic, server API and interface logic.

4.8 Deployment

The live application is currently lodged in university servers kindly provided to us by the Computer and Network Services at IST¹². It has a docker-compose configuration file orchestrating three docker containers, the postgres database, the back-end server, and the front-end container. The front-end container is based on a Nginx image and is responsible for receiving outside requests and delivering the Hypertext Markup Language (HTML)/Cascading Style Sheets (CSS)/JavaScript bundle or redirecting requests to the back-end container. In the front-end container there is an Nginx configuration file responsible for setting up Secure Sockets Layer (SSL) certificates used for Hypertext Transfer Protocol Secure (HTTPS) communication. To verify that this configuration provides adequate security we used SSLabs¹³ online tool (report available in Appendix B).

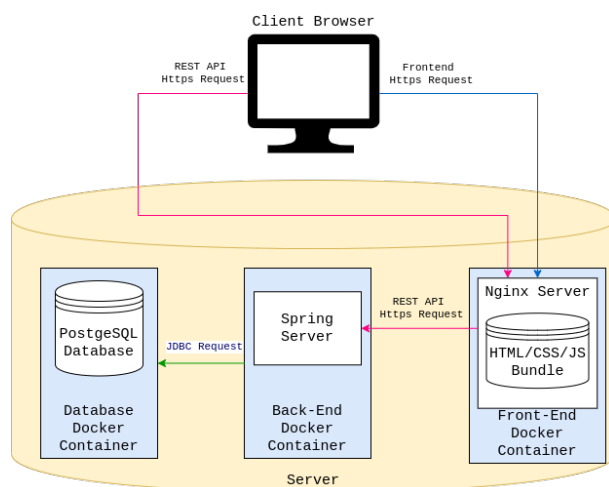


Figure 4.12: Interaction between client, server and containers

4.9 Continuous Integration

To make sure that the code submitted to the project repository is continuously integrated without breaking existing functionalities, Github Actions was configured to continuously run all tests on pull requests. Afterwards a code coverage report is uploaded to the Codecov¹⁴ dashboard which calculates code coverage changes to each module and adds comments on the Github pull request page. Similarly there is a Cypress dashboard that presents information about end-to-end tests ran with the pull request code. This information can be seen in a pull request like <https://github.com/socialsoftware/quizzes-tutor/pull/134>. This pull request was opened by Snyk¹⁵ an automated tool that checks if project dependencies are up to date and automatically opens pull requests to update them and prevent security issues with deprecated dependencies.

¹²<https://si.tecnico.ulisboa.pt/en/>

¹³<https://www.ssllabs.com/ssltest/analyze.html>

¹⁴<https://codecov.io/>

¹⁵<https://snyk.io/>

4.10 User Community

During the first semester of use in the Software Architecture course, students would occasionally give feedback regarding the system after class or by email to the teacher. However as our user base grew to incorporate other courses and other teachers we felt necessary to create a central way to communicate and obtain feedback from teachers. As such we created a [Slack Work-space](#), currently with 7 teachers which have already raised several usability issues like button sizes, date input, question correction strategy, etc that have helped the system reach their expectations and needs quicker.

4.11 Architectural Accomplishments

The developed system features and implementation meets the requirements mentioned in Section 3.6. Teachers and students have access to a web application with FenixEdu authentication that provides teachers with the ability to create and edit MCQs and quizzes as well as viewing quiz results and question difficulty, Students are able to solve assigned quizzes in the specified time frame and/or generate quizzes from the existing database to perform self-assessment activities. The following Chapter 5 will analyse the degree at which its execution and usage was able to accomplish the goals proposed in Section 1.1.

5

Result Analysis

Contents

5.1 Education Improvements	30
5.2 System Usage	33
5.3 Course Engagement and Assessment Improvements	34
5.4 Usability & Performance	36
5.5 Analysis Conclusion	37

The Quizzes Tutor system has been in use since September 20th, 2019 for courses of IST. During the first semester it was available to students of the Software Architecture course of the Information and Software Engineering Master's Degree. In this course the professor would make available 2 quizzes per week to students to study as optional self-assessment, with no evaluation being made using the system. During the second semester it was mainly used by the Software Engineering course of the Information and Software Engineering Bachelor's Degree, where it would continue to be used for self-assessment but additionally there were in-class quizzes that students could opt in to be a component in their evaluation and improve their final grade. To evaluate the success of this project during these two semesters, it is fundamental to define criteria. The four criteria we will use to evaluate the system, its implementation and usage in the context of the courses are the following:

1. Did the system improve student's knowledge?
2. Did students use the system and prefer it?
3. Did course engagement and assessment improve?
4. Was the system, usable and performant enough to meet its needs?

5.1 Education Improvements

It is a complex task to analytically compare student's knowledge, however most courses attribute a grade to students to represent their proficiency with the course's curriculum, therefore to evaluate this criteria we will use grades as an objective metric of proficiency. For this analysis, we'll focus on comparing the results from the Software Architecture course (first semester) with the previous years. In the context of this course the system was only used as an optional self-assessment tool with no grade component incentive or any other incentive for students to use the system. The reason for not using data from the Software Engineering course is due to the significant disruption of classes due to the COVID-19 pandemic both in classes and evaluation methodology that can prevent a reasonable comparison with previous years. Another advantage of the Software Architecture course is that it has been taught by the same professor for the last 10 years, so the curriculum, classes and evaluation differences over the years is minimized (as much as possible). The evaluation of the Software Architecture course is comprised of three components, an optional series of small mini-tests along the year that can be used to improve the final grade, the project grade and a final exam. If students fail the first exam they can still retry at a second exam. The mini-tests are comprised of 5 MCQs and exams have 20 MCQs and a written component. To more accurately compare students performance, we'll disregard penalties for wrong answers, only consider MCQs and will average results do disregard size of the exam. This process will also make it easier to use the graphs currently in the system since this information was already imported.

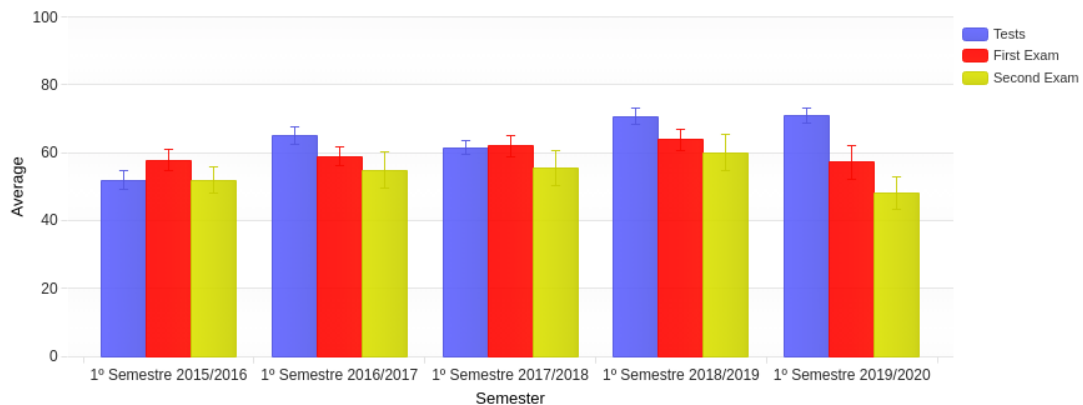


Figure 5.1: Average results from Software Architecture exams and mini-tests

In fig. 5.1 we can see the average grade of mini-tests, first exam and second exam from the previous years and from the current year 2019/2020. We can see that the average results from the mini-tests have been rising slightly and leveling this year, while the average score of MCQs in exams has dropped this year.

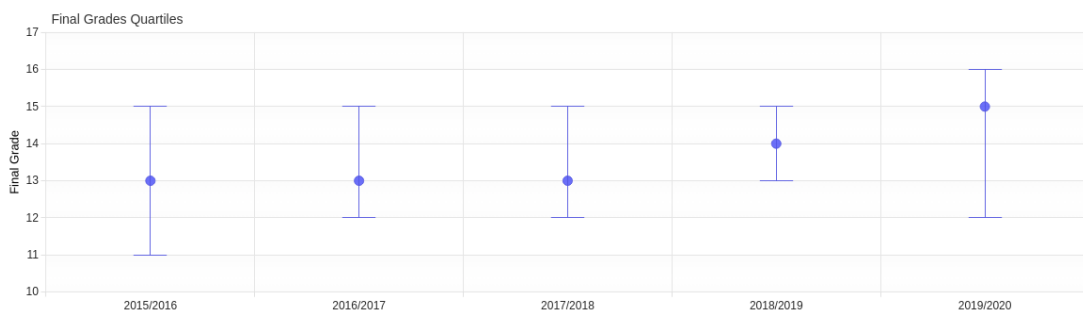


Figure 5.2: Final Grades Quartiles

Meanwhile, if we look at the final results, including all components of evaluation (Figure 5.2) we can actually see the opposite, meaning that grades have actually improved. In Figure 5.3 we can see more clearly that the percentage of students with grades 13-14 has decreased while students with 15 increased. This grade however is more dependant of evaluation methods and as such makes the comparison less reliable.

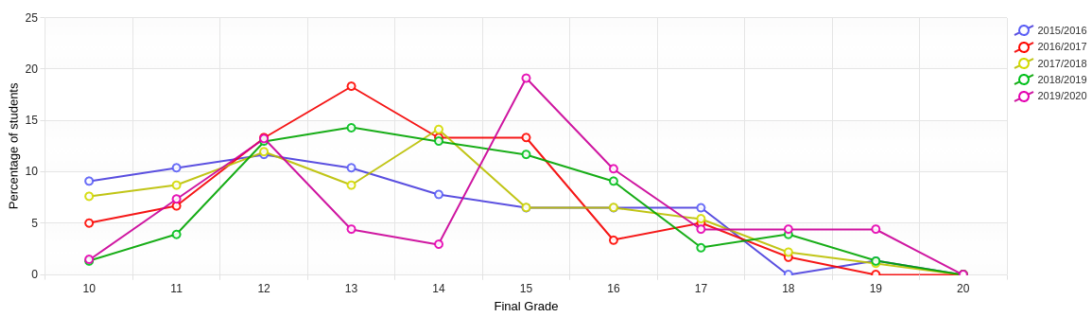


Figure 5.3: Final Grades

Finally, we want to investigate if these results can be linked with system usage or student performance in the system. Looking at Figure 5.4 where final course grade is related with the number of question answers made in the system we can see that the relation is very subtle. Looking at the relation between user proficiency in the system and final grade in Figure 5.5 we can see a slightly more significant relation between the two, meaning that students with good results in the system have better grades in the course.

Figure 5.4: Relation between usage and final grades

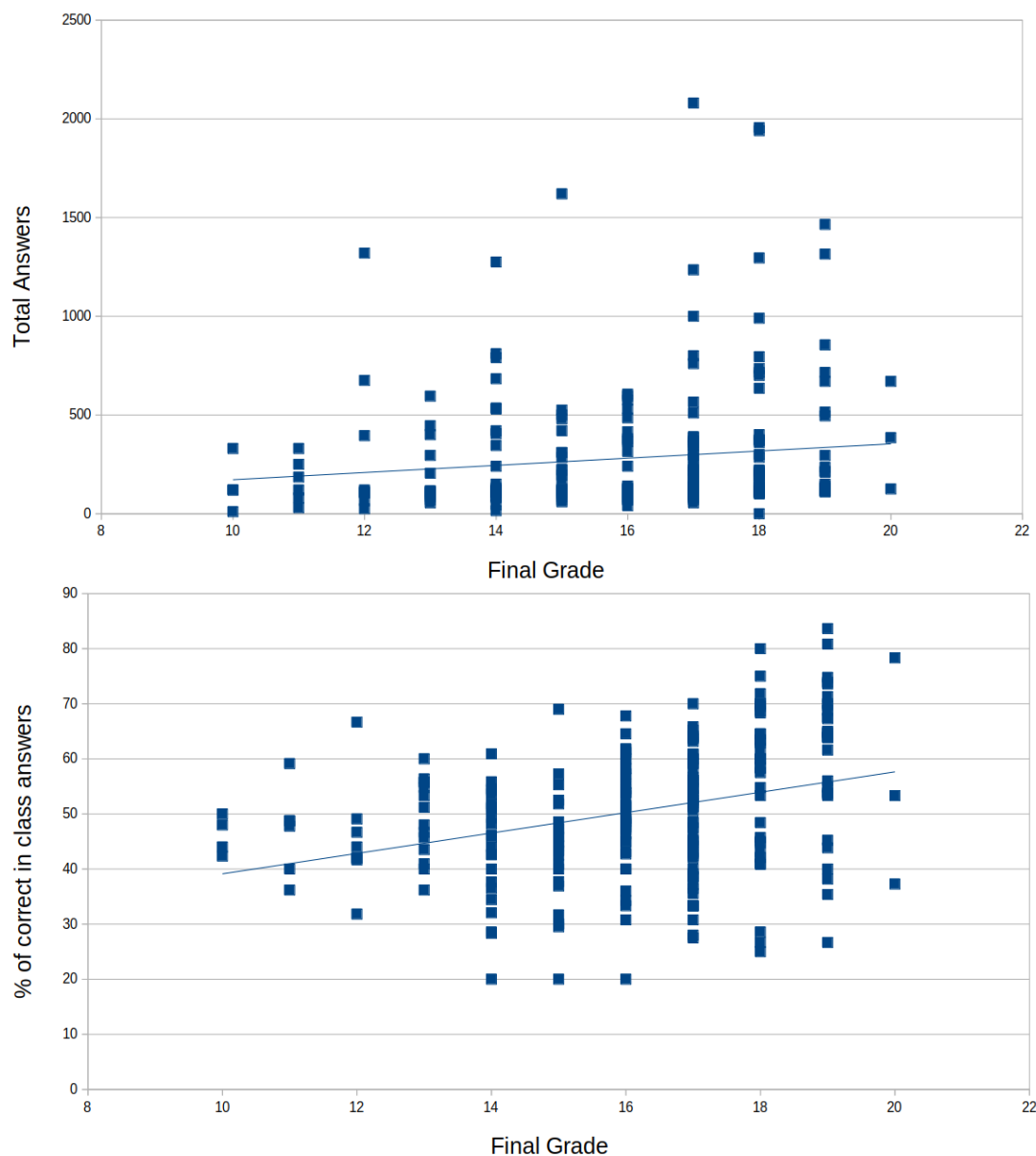


Figure 5.5: Relation between system proficiency and final grade

5.2 System Usage

From the system's perspective, its most important goal is to be used. So, to evaluate if students were using the platform, we had Nginx logs with each request to the server as well as a log table in the database that recorded simple operations during the first semester. In the end, this data proved unreliable due to inconsistencies between the two datasets, so instead we used the answer date of each quiz to establish how many quizzes were solved per day and evaluate the system usage based on that information. The graph in Figure 5.6 (available in the Quizzes Tutor for teachers to see), shows the data for both semesters for all courses using the system. From the 21st of September, 2019 to the 22nd of June, 2020 there was an average of 87 quizzes per day for a total of 16810 answers to quizzes. The graph shows a very spiky usage where each spike can be correlated with evaluations the following day or with quizzes made available by the teacher. The maximum answers per day were 1221 solved quizzes in the day before the Software Engineering exam. These were answered by 156 students, meaning that 57% of the students taking the exam (273) chose to use the system to study the day before the exam.

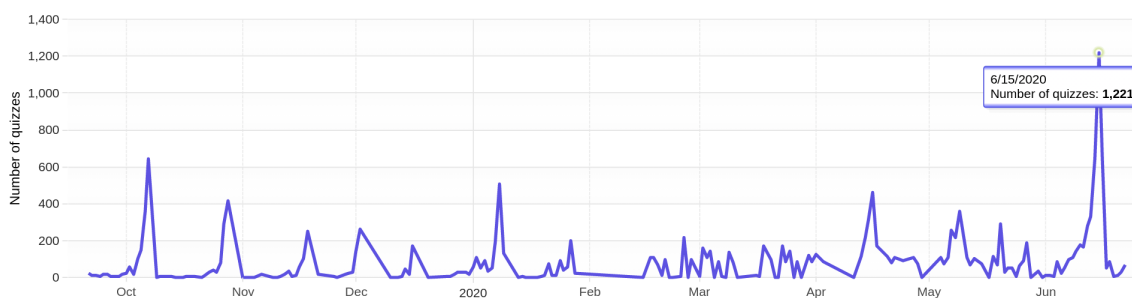


Figure 5.6: Solved quizzes per day

During the first semester from September to February, the platform was available to the 68 students enrolled in the Software Architecture course, where 59 of them used the system at least once. During the second semester from February to July the platform was made available to every teacher in IST but mainly used by Software Engineering students. Besides these 2 courses there were another 40 courses that have been activated in the platform, meaning that a teacher of that course has logged in and signed up to use the system for their course. There was at the time of writing a total of 90 teachers and 801 students that have logged in into the system. However many of those activations have not been followed up with actual use. We can see in Table 5.1 the courses that have at least 20 questions and 1 quiz created by the teacher. We can see that 6 other courses besides Software Architecture and Software Engineering have found the system useful for their evaluation needs. The influx of teachers trying the platform can be explained by the page created by the Sharing Remote Teaching and Research Experiences (SaRTRE)¹ forum at Instituto Superior Técnico, which was created to help teachers move

¹<https://sartre.tecnico.ulisboa.pt/metodologias-e-ferramentas/ferramentas/quizzes-tecnico/>

their evaluation remote during the COVID-19 pandemic.

Course	Questions	Generated quizzes	Teacher quizzes	Students	Teachers
Software Architecture	674	5152	28	59	1
Software Engineering	394	4417	73	314	5
Programming languages	128	234	37	57	2
Autonomous Agents and Multi-Agent Systems	72	15	19	174	3
Project Risk Evaluation and Management	20	10	1	76	1
Decision Support Models	30	1	8	80	4
Health Systems	63	0	5	71	3
Sensors and Actuators	50	0	2	27	1

Table 5.1: Courses actively using the system

Based on this data we can confidently say that the system was successful in being adopted. Students moved their self-assessment study from looking at previous exams and tests to using the platform and teachers have chosen to use the system to conduct their evaluation.

5.3 Course Engagement and Assessment Improvements

Course engagement is also difficult to analytically analyze. If we consider students dropping out or failing the course as a lack of engagement, we can see on Figure 5.7 that the percentage of approved students for the Software Architecture course has remained very steady with very few students dropping out. If we consider that engaged students are more likely to use the system, we can see from Section 5.2 that users were motivated to use the system. One observation made by the Software Architecture teacher was that during office hours, students would often bring screenshots of questions from the system, to request explanations and discuss the reasoning behind those questions. Likewise, with Software Engineering students, that used Slack to communicate with the teachers, after each evaluation it would be common for students to flock to Slack to discuss and try to understand the reasoning behind certain answers.

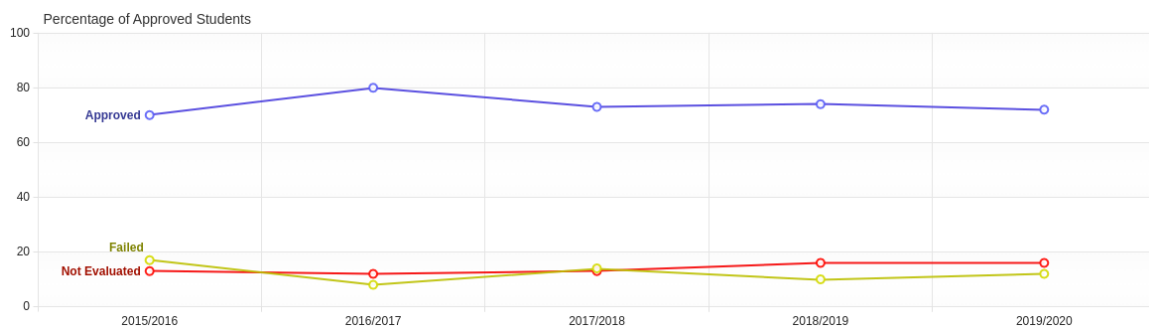


Figure 5.7: Percentage of approved students in Software Architecture

To better ascertain if course engagement and assessment had improved, at the end of both semesters, a questionnaire was made using Google Forms which obtained 27 answers from Software Architecture

students and 50 answers from Software Engineering students. The questionnaire included a section focused on general usability which is discussed in Section 5.4, a section focused on e-learning and self-assessment comprised of 5 questions devised based on the questionnaire described in [14] with simplifications and only applicable questions. The results are shown in Table 5.2 where the answers are given with a Likert Scale from 1 to 5. The results are clearly positive with students overwhelmingly liking the system and feeling that it was a helpful tool. The last part of the questionnaire was an open answer question asking for feedback as suggested in [15]. In this section, students requested new features, improvements they would like to see and pointed out small bugs meaning that students view continued development of the system as a necessity. One of the students took the opportunity to write "Cool Tool, thanks for that - it made the subject fun to learn".

Question	Average from 1-5	Standard Deviation
I was able to study effectively at any point of the day at my own pace	3.9	0.9
I think other courses should use this system because it is helpful to study	3.6	1.2
By using the system, I was able to understand what topics I needed to study more	3.5	1.2
I was able to effectively choose the topics I wanted to practise more	3.3	1.1
I felt motivated to study more after using the system	3.1	1.2

Table 5.2: Questionnaire results on self-assessment from 77 students

Question	Average from 1-5	Standard Deviation
Evaluation in each lecture helps me keep up with the course's subjects	4.2	0.9
I think optional evaluation with the system should continue to exist	3.9	1.2
Evaluation at the end of theoretical classes in the classroom is fair	3.7	1
Evaluation during classes makes me study more before the class	3.7	1.2
Evaluation at the end of theoretical classes at home is fair	3.5	1.1
Evaluation during classes makes me study more during the class	3.3	1.1
Evaluation during classes should replace exams	3.2	1.5

Table 5.3: Questionnaire results on assessment from 50 students

5.4 Usability & Performance

Being useful to students is a great advantage of the system but it does not mean that the system is well built. Therefore it is important to additionally evaluate usability and performance of the system. To evaluate usability, the questionnaire made to students included the System Usability Scale (SUS). SUS was created by John Brooke in 1986 and is considered the “quick and dirty” industry standard to evaluate the usability of any system from hardware devices and software interfaces to websites and yellow-pages. SUS is a reliable and valid measure of perceived usability and learnability that consists of 10 questions with a Likert Scale of 5 options which makes it incredibly easy, quick and cheap to answer and obtain results. The standard questions are:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Based on the 77 answers to these questions, the result was a score of 74/100. Since the average of this scale is 68, this puts this result in the 65 to 69 percentile, meaning that students found the system easy to use and to learn. It's worth mentioning that the 27 answers from Software Architecture course had a much higher result of 83.9/100 than the 50 answers from Software Engineering students that had a score of 68.7/100 even though their usage was practically identical.

To evaluate performance of the server application, several Jmeter tests were created. The following Table 5.4 is the result of one of those tests conducted on the live server. The results show that, using above average load, the average request to the server takes 1051 milliseconds to execute and the server is able to process 173 requests per second. These results are acceptable and a significant improvement compared to initial tests with previous versions of the back-end application.

Label	# Samples	Average	Min	Max	Error %	Throughput
Login as student	200	1553	185	6427	0.0	29.407
Get quizzes available	200	2896	366	6876	0.0	22.028
Start quiz	200	2251	305	6783	0.0	23.543
Submit answer	1000	511	12	4803	0.0	140.173
Conclude quiz	200	205	15	593	0.0	123.992
TOTAL	1800	1051	12	6876	0.0	173.276

Table 5.4: Jmeter results on live server testing student quiz answer

5.5 Analysis Conclusion

To conclude this section and looking back at the four established criteria, it is possible to affirm with a strong confidence that Item 2 was successfully accomplished with students usage being high and above expectations. Through the questionnaires we can also affirm that Item 3 and Item 4 objectives were met with students finding the system useful and usable and more important supporting continued development of the tool for future students. Lastly, Item 1 was inconclusive with more research being required with larger samples sizes and more comparable situations for it to be possible to be sustained.

6

Conclusion

Contents

6.1 System Limitations and Future Work	39
--	----

The Quizzes Tutor platform was originally devised by the Software Architecture professor to facilitate question management and sharing in the context of the IMPRESS project. This was a great opportunity to also provide students of the course with a tool for self-assessment using the those questions. As development progressed, it became a system for any course of IST to perform any assessment activity based on MCQs and its adoption was fairly successful with the development of new features currently under way. This thesis described the development of the project including a description of pedagogical principles related with the use of MCQs and E-assessment platforms, description of other popular LMS systems with an analysis of their features, description of the existing functionalities and finally an evaluation of its adoption. This project was carried out with the hope of improving student's feedback, improving their learning process and motivation as well as helping teachers improve their courses and perform their assessment activities. It is also the hope behind this thesis to inspire the conception and adoption of similar systems to improve education and promote self-assessment.

6.1 System Limitations and Future Work

This section holds a list of functionalities that are currently being added by other developers and students or that are can be useful in other future development phases.

6.1.1 Features from the Software Engineering Project

As previously mentioned, Software Engineering students in addition to using the system for assessment activities have also used its codebase as basis for their course project. Students were divided in groups of 9 which in turn were sub-divided in three groups of 2 with each sub-group developing a different feature. The 3 predetermined features were the implementation of a question suggestion mechanism that allows students to propose new questions to the teacher. This is an activity suggested in the first principle of [8] that makes students apply their knowledge in a way that makes them consider the goal of the evaluation. The second feature is the explanation request, where students can ask for clarification to the teacher after solving a question, a very requested feature in the Software Architecture questionnaire. The last feature was the implementation of a tournament style quiz where student would create a quiz and solve it with other users, adding a bit of gamification to the system. Overall, students were very pleased to have a real world example to work with, especially one where they were also the users of the system. Once the semester was over, a few students were selected to incorporate their features into the system.

6.1.2 Question Type Expansion

The existing system was created to leverage the existing question dataset in the MCQ format which also provides many advantages as described in this thesis. However, diversity of question types would allow the system to better adapt to different use cases of other teachers and evaluate different skills that MCQ are unable to ascertain. Expanding question types is the focus of another Master's thesis currently taking place that would allow multiple question types with a focus on questions for Software Engineering that allow text selection from a snippet of code.

6.1.3 Fraud detection

While the system was used in classes, a QR code mechanism was implemented to prevent students outside the class to perform the quiz. This solution isn't perfect because it led to screenshots of the QR code being sent by messages to students outside the classroom. As a secondary validation, the professor had to ask the last students to join the quiz to validate their presence with him personally, which is time consuming and not 100% effective but good enough at catching these cases and provides another level of security. This type of fraud detection is important in a system that is used to determine students grades so there is currently another Master's student working on additional mechanisms of fraud detection.

6.1.4 Infrastructure

As the system becomes more complex and the number of users increases, it is imperative that the servers remain reliable and fault tolerant. Therefore one of the areas that can be improved is the monitoring of the servers resources (CPU, memory) and provide that information to the administrator of the server. Improving the current server with a load balancer and adding dynamic server deployment based on usage can also be helpful but requires considerable changes to a distributed database. Other related issues that can be improved are spam prevention, automatic SSL certificates renewal, Google analytics integration and true continuous deployment from a github.

6.1.5 Natural Language Processing and Machine Learning

The Software Architecture and Software Engineering courses have reached a good amount of questions in the database that can be analysed using a Natural Language Processing (NLP) library such as CoreNLP¹ or Opennlp². A useful feature of these libraries is the document categorizer or document classification that could be useful to suggest topics while teachers create their questions. Another possible use of Artificial Intelligence in this project, would be to improve the generation of the quizzes. Taking into account other students usage and previous user answers, a library like Deeplearning4j³ could improve the quiz generation and choose questions more likely for the student to get wrong.

¹<https://stanfordnlp.github.io/CoreNLP/>

²<https://opennlp.apache.org/>

³<https://deeplearning4j.org/index.html>

Bibliography

- [1] J. Bernardo Rocha, L. Felipe Coimbra Costa, R. Prada, A. Rito Silva, D. Gonçalves, and P. Correia, "Quizzes (as self-regulated learning) in software engineering education (forthcoming)," in *Proceedings of CSEET 2020*. IEEE, 2020.
- [2] A. Sangrà, D. Vlachopoulos, and N. Cabrera, "Building an inclusive definition of e-learning: An approach to the conceptual framework," *International Review of Research in Open and Distributed Learning*, vol. 13, no. 2, pp. 145–159, 2012.
- [3] W. Watson and S. L. Watson, "An argument for clarity: What are learning management systems, what are they not, and what should they become." 2007.
- [4] N. Alruwais, G. Wills, and M. Wald, "Advantages and challenges of using e-assessment," *International Journal of Information and Education Technology*, vol. 8, no. 1, pp. 34–37, 2018.
- [5] S. Jordan, "E-assessment: Past, present and future," *New Directions in the Teaching of Physical Sciences*, no. 9, pp. 87–106, 2013.
- [6] A. Adams and G. Clough, "The e-assessment burger: supporting the before and after in e-assessment systems," *Interaction Design and Architecture (s)*, vol. 25, pp. 39–57, 2015.
- [7] K. Scouller, "The influence of assessment method on students' learning approaches: Multiple choice question examination versus assignment essay," *Higher Education*, vol. 35, no. 4, pp. 453–472, 1998.
- [8] D. Nicol, "E-assessment by design: using multiple-choice tests to good effect," *Journal of Further and higher Education*, vol. 31, no. 1, pp. 53–64, 2007.
- [9] J. Considine, M. Botti, and S. Thomas, "Design, format, validity and reliability of multiple choice questions for use in nursing research and education," *Collegian*, vol. 12, no. 1, pp. 19–24, 2005.
- [10] K. Brandl, "Review of are you ready to" moodle"?" *Language Learning & Technology*, vol. 9, no. 2, pp. 16–23, 2005.

- [11] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Engaging engineering students with gamification," in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*. IEEE, 2013, pp. 1–8.
- [12] J. Bull and M. Danson, "Computer assisted assessment (caa)(york, learning and teaching support network)," 2004.
- [13] A. Gardner-Medwin, "12 confidence-based marking," *Innovative assessment in higher education*, p. 141, 2006.
- [14] P. Zaharias and A. Poylymenakou, "Developing a usability evaluation method for e-learning applications: Beyond functional usability," *Intl. Journal of Human–Computer Interaction*, vol. 25, no. 1, pp. 75–98, 2009.
- [15] W. T. Nakamura, E. H. T. de Oliveira, and T. Conte, "Usability and user experience evaluation of learning management systems-a systematic mapping study," in *International Conference on Enterprise Information Systems*, vol. 2. SCITEPRESS, 2017, pp. 97–108.



Sample Code Snippets

In this appendix, a few shortened snippets of code are presented to demonstrate the typical transfer of data from the database, Spring Service, Spring Controller to the User Interface. In listing A.1 there is a JPA entity for the question with the necessary annotations. In the domain it is important to clear state the relations with other entities, in this case there is a one to many relation with the entity Option, where one question can have multiple options.

Listing A.1: JPA Question Domain Entity

```
1 @Entity
2 @Table(name = "questions")
3 public class Question implements DomainEntity {
4     @Id
5     @GeneratedValue(strategy = GenerationType.IDENTITY)
6     private Integer id;
7
8     @Column(columnDefinition = "TEXT")
9     private String content;
10
11     @OneToMany(cascade = CascadeType.ALL, mappedBy = "question", fetch =
        FetchType.EAGER, orphanRemoval=true)
12     private final List<Option> options = new ArrayList<>();
13
14
15     public Question() {
16     }
17 }
```

In listing A.2 there is a Spring Service method with the transactional annotation. Repositories are autowired making them accessible in the method to obtain, create and delete data. It is also possible to create new SQL queries in the Repository class to be used in the services.

Listing A.2: Spring Service method to create Question

```
1 @Service
2 public class QuestionService {
3
4     @Autowired
5     private CourseRepository courseRepository;
6
7     @Autowired
8     private QuestionRepository questionRepository;
9
10    @Retryable(
11        value = { SQLException.class },
12        backoff = @Backoff(delay = 5000))
13    @Transactional(isolation = Isolation.READ_COMMITTED)
14    public QuestionDto createQuestion(int courseId, QuestionDto questionDto)
15    {
16        Course course = courseRepository.findById(courseId)
17            .orElseThrow(() -> new TutorException(COURSE_NOT_FOUND, courseId)
18            );
19
20        Question question = new Question(course, questionDto);
21        questionRepository.save(question);
22        return new QuestionDto(question);
23    }
24 }
```

In listing A.3 there is the API specification with the Mapping annotation and the access control annotation PreAuthorize that verifies if the user has the necessary credentials to make the request. The methods in the the controller then call the Service which is also autowired and return a Data Transfer Object seen in listing A.4.

Listing A.3: Spring Controller method to create Question

```
1 @RestController
2 public class QuestionController {
3     @Autowired
4     private QuestionService questionService;
5
6     @PostMapping("/courses/{courseId}/questions")
7     @PreAuthorize("hasRole('ROLE_TEACHER') and hasPermission(#courseId, '
8         COURSE.ACCESS')")
9     public QuestionDto createQuestion(@PathVariable int courseId, @Valid
10        @RequestBody QuestionDto question) {
11        question.setStatus(Question.Status.AVAILABLE.name());
12        return this.questionService.createQuestion(courseId, question);
13    }
14 }
```

Listing A.4: Question Data Transfer Object (DTO)

```
1 public class QuestionDto implements Serializable {
2     private Integer id;
3     private List<OptionDto> options = new ArrayList<>();
4
5     public QuestionDto() {
6     }
7
8     public QuestionDto(Question question) {
9         this.id = question.getId();
10        this.options = question.getOptions().stream().map(OptionDto::new).
11            collect(Collectors.toList());
12    }
13
14    public Integer getId() {
15        return id;
16    }
17
18    public List<OptionDto> getOptions() {
19        return options;
20    }
21
22    public void setOptions(List<OptionDto> options) {
23        this.options = options;
24    }
25 }
```

On the front-end code seen in listing A.5 the Axios library is used to create an httpClient variable which is used to make the requests to the back-end applications. In this snippet of code it is also possible to see the use of Promises to keep the code executing asynchronously improving performance and the use of a store to keep information available in to multiple views.

Listing A.5: Axios http request

```
1 static async createQuestion(question: Question): Promise<Question> {
2     return httpClient
3         .post(
4             `/courses/${Store.getters.getCurrentCourse.courseId}/questions/`,
5             question
6         )
7         .then(response => {
8             return new Question(response.data);
9         })
10        .catch(async error => {
11            throw Error(await this.errorMessage(error));
12        });
13 }
```

Information from the back-end is then converted into a JavaScript model object seen in listing A.6 with full TypeScript support to be easily used in the view.

Listing A.6: Question frontend Model

```
1 export default class Question {
2   id: number | null = null;
3   options: Option[] = [new Option(), new Option(), new Option(), new Option()
4     ];
5   constructor(jsonObj?: Question) {
6     if (jsonObj) {
7       this.id = jsonObj.id;
8
9       this.options = jsonObj.options.map(
10         (option: Option) => new Option(option)
11       );
12     }
13   }
14 }
```

Finally, in listing A.7 there is a Vue component which is divided in template with the HTML code and the TypeScript decorator for class-style Vue component where variables and methods are declared. This particular component receives a Question as a Prop and shows it to the user.

Listing A.7: Vue Component to show Question

```
1 <template>
2   <div>
3     <span v-html="convertMarkdown(question.content, question.image)" />
4     <br />
5     <ul>
6       <li v-for="option in question.options" :key="option.number">
7         <span
8           v-if="option.correct"
9           v-html="convertMarkdown('*{0+FFF}* ' + option.content)"
10          v-bind:class="[option.correct ? 'font-weight-bold' : '']"
11         />
12         <span v-else v-html="convertMarkdown(option.content)" />
13       </li>
14     </ul>
15     <br />
16   </div>
17 </template>
18
19 <script lang="ts">
20 import { Component, Vue, Prop } from 'vue-property-decorator';
21 import { convertMarkdown } from '@services/ConvertMarkdownService';
22 import Question from '@models/management/Question';
23 import Image from '@models/management/Image';
24
25 @Component
26 export default class ShowQuestion extends Vue {
```

```
27  @Prop({ type: Question, required: true }) readonly question!: Question;
28
29  convertMarkdown(text: string, image: Image | null = null): string {
30    return convertMarkdown(text, image);
31  }
32 }
33 </script>
```



SSL Configuration Report

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > quizzes-tutor.tecnico.ulisboa.pt

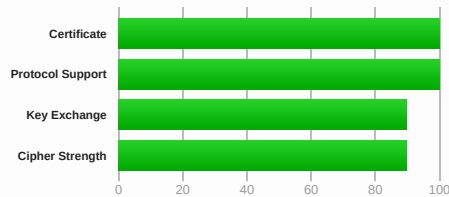
SSL Report: quizzes-tutor.tecnico.ulisboa.pt (192.92.147.23)

Assessed on: Thu, 25 Jun 2020 16:14:44 UTC | [Hide](#) | [Clear cache](#)

[Scan Another »](#)

Summary

Overall Rating



Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. [MORE INFO »](#)

Certificate #1: RSA 4096 bits (SHA512withRSA)



Server Key and Certificate #1



Subject	quizzes-tutor.tecnico.ulisboa.pt Fingerprint SHA256: b3787711e0c45323c8229e57054ef6f294befb3860c3d49db3b9407d7bb26799 Pin SHA256: pJUAQjKXqH2jFXIEFKJ76dqlwu96TmWBUjmyzSfGK8=
Common names	quizzes-tutor.tecnico.ulisboa.pt
Alternative names	quizzes-tutor.tecnico.ulisboa.pt www.quizzes-tutor.tecnico.ulisboa.pt
Serial Number	0b88fdca5d819a57193c16937b149e7b
Valid from	Tue, 22 Oct 2019 00:00:00 UTC
Valid until	Tue, 26 Oct 2021 12:00:00 UTC (expires in 1 year and 4 months)
Key	RSA 4096 bits (e 65537)
Weak key (Debian)	No
Issuer	TERENA SSL CA 3 AIA: http://cacerts.digicert.com/TERENASSLCA3.crt
Signature algorithm	SHA512withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl3.digicert.com/TERENASSLCA3.crl OCSP: http://ocsp.digicert.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows



Additional Certificates (if supplied)



Certificates provided	3 (4319 bytes)
Chain issues	Contains anchor

#2

Additional Certificates (if supplied)

Subject

TERENA SSL CA 3

Fingerprint SHA256: beb9efe9b1a73c841b375a90e5ff8048848e3a2af66f6c4dd7b938d6fe8c5d8

Pin SHA256: 8651wEkMkH5ftiaLp57oqmx3KHTFzDgp7ZeJXR0ToBs=

Valid until

Mon, 18 Nov 2024 12:00:00 UTC (expires in 4 years and 4 months)

Key

RSA 2048 bits (e 65537)

Issuer

DigiCert Assured ID Root CA

Signature algorithm

SHA256withRSA

#3

Subject

DigiCert Assured ID Root CA [In trust store](#)

Fingerprint SHA256: 3e9099b5015e8f486c00bcea9d11ee721faba355a89bc1df69561e3dc6325c

Pin SHA256: /L/tz7ekCWanjDOCv5EqXls2IOaThEAOH2Bg4BT/o=

Valid until

Mon, 10 Nov 2031 00:00:00 UTC (expires in 11 years and 4 months)

Key

RSA 2048 bits (e 65537)

Issuer

DigiCert Assured ID Root CA Self-signed

Signature algorithm

SHA1withRSA Weak, but no impact on root certificate

Link icon

Certification Paths

Link icon

Click here to expand

Configuration

Protocols

TLS 1.3

No

TLS 1.2

Yes

TLS 1.1

No

TLS 1.0

No

SSL 3

No

SSL 2

No

Cipher Suites

TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)

ECDH x25519 (eq. 3072 bits RSA)

FS

256

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

ECDH x25519 (eq. 3072 bits RSA)

FS

128

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK256

TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (0xc077)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK256

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK128

TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0xc076)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK128

TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)

ECDH x25519 (eq. 3072 bits RSA)

FS

256

TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 (0xc061)

ECDH x25519 (eq. 3072 bits RSA)

FS

256

TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 (0xc060)

ECDH x25519 (eq. 3072 bits RSA)

FS

128

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK256

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)

ECDH x25519 (eq. 3072 bits RSA)

FS

WEAK128

Handshake Simulation

Android 4.4.2

RSA 4096 (SHA512)

TLS 1.2

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

ECDH secp256r1

FS

Android 5.0.0

RSA 4096 (SHA512)

TLS 1.2

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

ECDH secp256r1

FS

Android 6.0

RSA 4096 (SHA512)

TLS 1.2 > http/1.1

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

ECDH secp256r1

FS

Android 8.0

RSA 4096 (SHA512)

TLS 1.2 > h2

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

ECDH x25519

FS

Android 8.0

RSA 4096 (SHA512)

TLS 1.2 > h2

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

ECDH x25519

FS

Android 8.1

RSA 4096 (SHA512)

TLS 1.2 > h2

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

ECDH x25519

FS

51

Handshake Simulation

Android 9.0	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
BingPreview Jan 2015	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Chrome 49 / XP SP3	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Chrome 70 / Win 10	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Chrome 80 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Firefox 73 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Googlebot Feb 2018	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
IE 11 / Win 7 R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
IE 11 / Win 8.1 R	RSA 4096 (SHA512)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
IE 11 / Win Phone 8.1 R	RSA 4096 (SHA512)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDH secp256r1 FS
IE 11 / Win Phone 8.1 Update R	RSA 4096 (SHA512)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
IE 11 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 16 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 18 / Win 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java 8u161	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java 11.0.3	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java 12.0.1	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.1l R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.2s R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.1.0k R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
OpenSSL 1.1.1c R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Safari 6 / iOS 6.0.1	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
Safari 7 / iOS 7.1 R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDH secp256r1 FS
Safari 9 / iOS 9 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 12.1.2 / MacOS 10.14.6 Beta R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Safari 12.1.1 / iOS 12.3.1 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Apple ATS 9 / iOS 9 R	RSA 4096 (SHA512)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Yahoo Slurp Jan 2015	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
YandexBot Jan 2015	RSA 4096 (SHA512)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS

Not simulated clients (Protocol mismatch)



[Click here to expand](#)

- (1) Clients that do not support Forward Secrecy (FS) are excluded when determining support for it.
- (2) No support for virtual SSL hosting (SNI). Connects to the default site if the server uses SNI.
- (3) Only first connection attempt simulated. Browsers sometimes retry with a lower protocol version.
- (R) Denotes a reference browser or client, with which we expect better effective security.
- (All) We use defaults, but some platforms do not use their best protocols and features (e.g., Java 6 & 7, older IE).
- (All) Certificate trust is not checked in handshake simulation, we only perform TLS handshake.

Protocol Details



Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete	
Secure Renegotiation	Supported	
Secure Client-Initiated Renegotiation	No	
Insecure Client-Initiated Renegotiation	No	
BEAST attack	Mitigated server-side (more info)	
POODLE (SSLv3)	No, SSL 3 not supported (more info)	
POODLE (TLS)	No (more info)	
Zombie POODLE	No (more info)	TLS 1.2 : 0xc027
GOLDENDOODLE	No (more info)	TLS 1.2 : 0xc027
OpenSSL 0-Length	No (more info)	TLS 1.2 : 0xc027
Sleeping POODLE	No (more info)	TLS 1.2 : 0xc027
Downgrade attack prevention	Unknown (requires support for at least two protocols, excl. SSL2)	
SSL/TLS compression	No	
RC4	No	
Heartbeat (extension)	No	
Heartbleed (vulnerability)	No (more info)	
Ticketbleed (vulnerability)	No (more info)	
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)	
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)	
ROBOT (vulnerability)	No (more info)	
Forward Secrecy	Yes (with most browsers) ROBUST (more info)	
ALPN	Yes	h2 http/1.1
NPN	Yes	h2 http/1.1
Session resumption (caching)	No (IDs assigned but not accepted)	
Session resumption (tickets)	Yes	
OCSF stapling	No	
Strict Transport Security (HSTS)	Yes max-age=31536000; includeSubDomains	
HSTS Preloading	Not in: Chrome Edge Firefox IE	
Public Key Pinning (HPKP)	No (more info)	
Public Key Pinning Report-Only	No	
Public Key Pinning (Static)	No (more info)	
Long handshake intolerance	No	
TLS extension intolerance	No	
TLS version intolerance	No	
Incorrect SNI alerts	No	
Uses common DH primes	No, DHE suites not supported	
DH public server param (Ys) reuse	No, DHE suites not supported	
ECDH public server param reuse	No	
Supported Named Groups	x25519, secp256r1, x448, secp521r1, secp384r1 (server preferred order)	
SSL 2 handshake compatibility	No	



HTTP Requests



1 <https://quizzes-tutor.tecnico.ulisboa.pt/> (HTTP/1.1 200 OK)



Miscellaneous

Test date	Thu, 25 Jun 2020 16:13:13 UTC
Test duration	91.299 seconds
HTTP status code	200
HTTP server signature	nginx/1.18.0
Server hostname	rt-717508-t1.vps.tecnico.ulisboa.pt

SSL Report v2.1.5

Copyright © 2009-2020 [Qualys, Inc.](#) All Rights Reserved.

[Terms and Conditions](#)

[Try Qualys for free!](#) Experience the award-winning [Qualys Cloud Platform](#) and the entire collection of [Qualys Cloud Apps](#), including [certificate security](#) solutions.



Additional contributions

Article "Quizzes (as self-regulated learning) in Software Engineering Education" by José Bernardo Rocha, Luis Felipe Coimbra Costa, Rui Prada, António Rito Silva, Daniel Gonçalves, and Pedro Correia currently in proceedings of CSEET conference to be presented in November 2020.