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

Pedro Miguel Póvoa Correia
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
pedro.miguel.correia@tecnico.ulisboa.pt

***Abstract*—This document 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.**

***Index Terms*—E-assessment; Multiple Choice Questions**

I. Introduction

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 (MCQ) 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 (LMS) that help students in this regard are sometimes cumbersome to setup and use due to complex interfaces which discourage teachers from using them.

This work was developed in the context of the IMPRESS project funded by EU Erasmus+ Programme and Pedagogical Innovation Projects of Instituto Superior Técnico

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 document 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 [14] "in press" 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".

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?

II. 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 [1]. Directly from E-learning emerged the concept of LMS [2] 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 [3]. 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;

A. 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 [3] and [4] 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 [5] 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.

B. 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 [6] [7] [8] on the advantages and disadvantages of their use, which are hereafter succinctly presented.

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) MCQs Disadvantages: As discussed in [4], 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 reveling 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;

III. Feedback Principles

Since feedback is the most important component of formative assessment, responsible for promoting learning and the dispelling of misconceptions, seven feedback principles [7] 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;

IV. Project development

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 postgreSQL database.

A. Functionalities

The current functionalities can be seen live in https://quizzes-tutor.tecnico.ulisboa.pt/ in a demo environment or in a video demonstration[1] prepared by the Software Engineering teacher António Rito Silva. For the sake of completeness a list of features with corresponding screenshots are listed below:

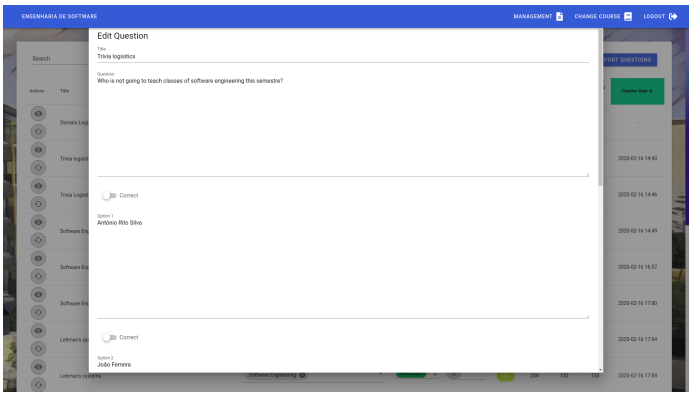

Fig. 1. Teacher editing a question

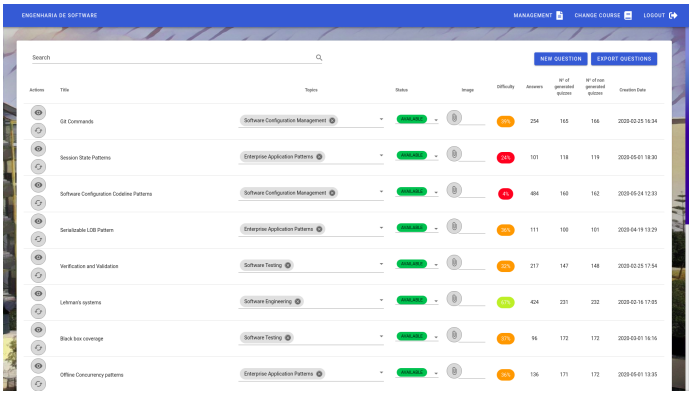

Fig. 2. Teacher's list of questions

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 fig:teacher-editing-question with full markdown support to add style, images, quotations, etc. In the question listing (fig:teacher-question-listing) each question can then 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 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 (fig:teacher-editing-assessment), 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.

Besides randomly generated quizzes based on assessments, teachers can also create quizzes with specific questions. In the quiz edition view, 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, so that only students with the QR code can start the quiz, 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 fig:student-answers) which can then be exported to 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 fig:tutor-students).

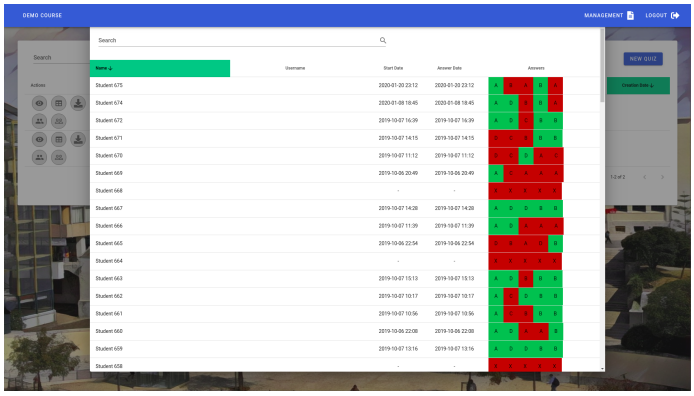

Fig. 3. Teacher's list of student answers for a quiz

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

[1]https://www.youtube.com/watch?v=PUDzoCqMznE

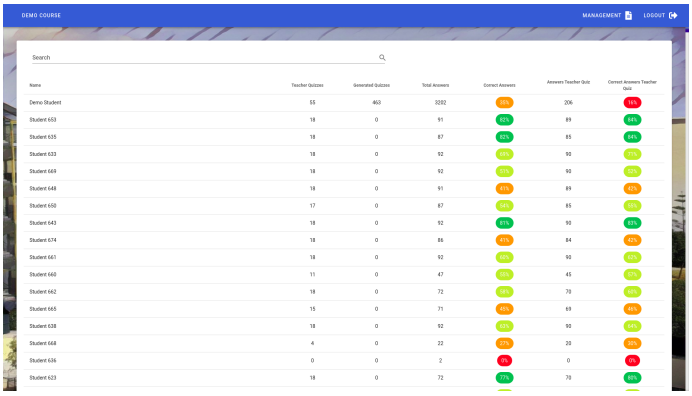

Fig. 4. Teacher's list of students

to the device's webcam to read a QR code. Once a student has started a quiz as seen in fig:student-quiz, 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.

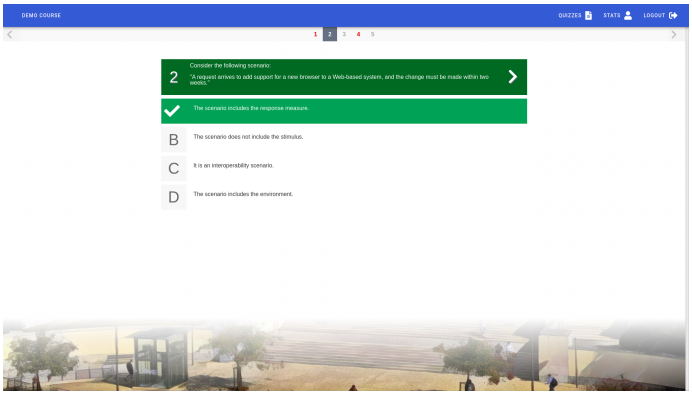

Fig. 5. Student answering quiz and results

Result Analysis

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?

V. 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. In fig:result-comparison 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.

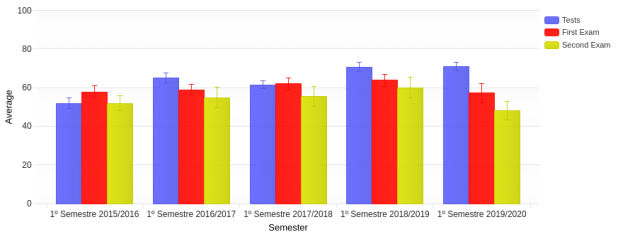

Fig. 6. Average results from Software Architecture exams and mini-tests

Meanwhile, if we look at the final results, including all components of evaluation (fig:final-grades) we can actually see the opposite, meaning that grades have actually improved. In fig:final-grades-lines 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.

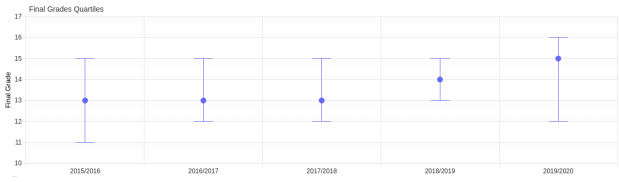

Fig. 7. Final Grades Quartiles

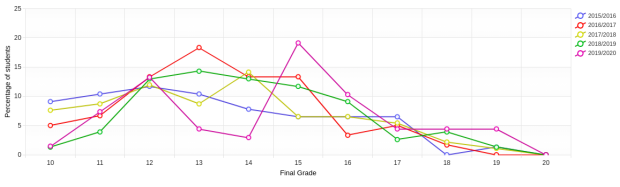

Fig. 8. Final Grades

Finally, we want to investigate if these results can be linked with system usage or student performance in the system. Looking at fig:answers-grades 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 fig:percentage-grades 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.

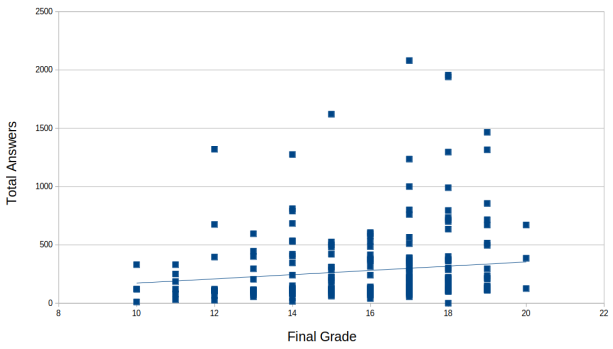

Fig. 9. Relation between usage and final grades

VI. 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

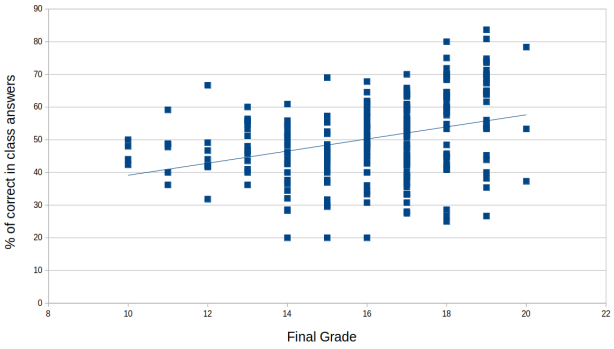

Fig. 10. Relation between system proficiency and final grade

table in the database that recorded simple operations during the first semester. In the end, this data proved unreliable, 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 fig:daily-quizzes (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 were 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.

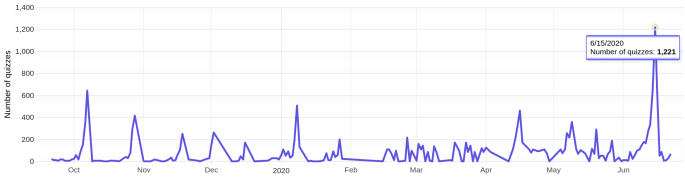

Fig. 11. Solved quizzes per day

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 tab:courses 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)[2] forum at Instituto Superior Técnico, which was created to help teachers move 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 I
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.

VII. 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 fig:approved-students 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 sec:system-usage 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.

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

[2]https://sartre.tecnico.ulisboa.pt/metodologias-e-ferramentas/ferramentas/quizzes-tecnico/

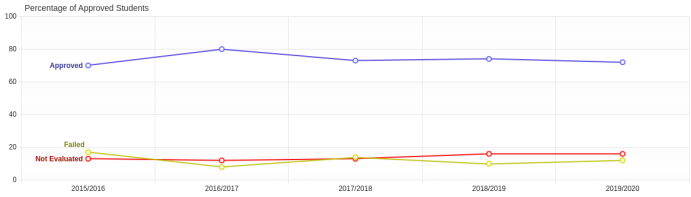

Fig. 12. Percentage of approved students in Software Architecture

a section focused on general usability which is discussed in sec:usability-performance, a section focused on e-learning and self-assessment comprised of 5 questions devised based on the questionnaire described in [12] with simplifications and only applicable questions. The results are shown in tab:questionnaire-self-assessment 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 [13]. 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 II
QUESTIONNAIRE RESULTS ON SELF-ASSESSMENT FROM 77 STUDENTS

VIII. USABILITY

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 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:

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 III
QUESTIONNAIRE RESULTS ON ASSESSMENT FROM 50 STUDENTS

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.

IX. CONCLUSION

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 feedback principles related with the use of MCQs and E-assessment platforms, 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.

X. 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.

A. 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 [7] 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.

B. 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.

C. 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.

D. 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.

ACKNOWLEDGMENT

A special thank you to 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.

REFERENCES

[1] Sangrà, A., Vlachopoulos, D., & Cabrera, N. (2012). Building an inclusive definition of e-learning: An approach to the conceptual framework. The International Review of Research in Open and Distributed Learning, 13(2), 145-159.

[2] Watson, W., & Watson, S. L. (2007). An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become.

[3] Alruwais, N., Wills, G., & Wald, M. (2018). Advantages and Challenges of Using E-assessment. International Journal of Information and Education Technology (IJIET), 8(1), 34-37.

[4] Jordan, S. (2016). E-assessment: Past, present and future. New Directions in the Teaching of Physical Sciences, 0(9), 87-106. doi:https://doi.org/10.29311/ndtps.v0i9.504

[5] Adams, Anne and Clough, Gill (2015). The e-assessment burger: supporting the before and after in e-assessment systems. Interaction Design and Architecture(s)(25) pp. 39–57.

[6] Scouller, K. (1998). The influence of assessment method on students' learning approaches: Multiple choice question examination versus assignment essay. Higher Education, 35(4), 453-472.

[7] David Nicol (2007) E-assessment by design using multiple-choice tests to good effect, Journal of Further and Higher Education, 31:1, 53-64, DOI: 10.1080/03098770601167922

[8] Considine, J., Botti, M., & Thomas, S. (2005). Design, format, validity and reliability of multiple choice questions for use in nursing research and education. Collegian, 12(1), 19-24.

[9] Brandl, K. (2005). Are you ready to "Moodle". Language Learning & Technology, 9(2), 16-23.

[10] Bull, J.& Danson, M. (2004) Computer assisted assessment (CAA) (York, Learning and Teaching Support Network).

[11] Gardner-Medwin, A. R. (2006) Confidence-based marking: towards deeper learning and better exams, in: C. Bryan & K. Clegg (Eds) Innovative assessment in higher education (London, Taylor & Francis).

[12] Zaharias, Panagiotis, and Angeliki Poylymenakou. "Developing a usability evaluation method for e-learning applications: Beyond functional usability." Intl. Journal of Human–Computer Interaction 25.1 (2009): 75-98.

[13] Nakamura, Walter Takashi, Elaine Harada Teixeira de Oliveira, and Tayana Conte. "Usability and User Experience Evaluation of Learning Management Systems-A Systematic Mapping Study." International Conference on Enterprise Information Systems. Vol. 2. SCITEPRESS, 2017.

[14] Bernardo Rocha, José, Felipe Coimbra Costa, Luis, Prada, Rui, Rito Silva, António, Gonçalves, Daniel and Correia, Pedro. "Quizzes (as self-regulated learning) in Software Engineering Education (Forthcoming)", Procedings of CSEET, 2020