2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)

Instructor Support Module in a Web-Based Peer Assessment Platform

Gabriel Badea
Computers and Information Technology Department
University of Craiova
Craiova, Romania
gabriel.badea@yahoo.com

Elvira Popescu
Computers and Information Technology Department
University of Craiova
Craiova, Romania
popescu_elvira@software.ucv.ro

***Abstract*—Benefits of the peer assessment for both teachers and students are well documented in the literature. From the instructor perspective, it can scale grading in large classes, track students' progress, reduce the grading burden and provide an estimation of the final course grade. The existing peer assessment platforms generally offer limited support for the teacher; many of them are based on a predefined peer assessment scenario and the instructor does not have the opportunity to configure the workflow, review allocation or grade computing mechanisms. In this context, we propose a peer assessment system, called LearnEval, which provides comprehensive instructor support: creation and editing of assignments, detailed monitoring of student activity, including various scores and statistics, learner model visualization; in addition, the teacher can configure the scenario and scores in order to fit the needs of the course. The paper describes the LearnEval instructor module in terms of functionalities, implementation, as well as a pilot study in the context of a Multimedia Technologies in E-Learning project.**

Keywords—technology-enhanced assessment, teacher support, peer evaluation

I. INTRODUCTION

Peer assessment has been successfully applied by instructors in various pedagogical scenarios [3, 9]. Traditionally, the process was very time-consuming for the teachers, as they had to manage paper-based reviews and process them by hand. In recent years, several peer assessment platforms have been proposed in the literature [1, 7, 8, 10]. Generally, these platforms allow the instructor to manage the process of defining assignments, evaluating the student work and visualizing the peer review data, but few platforms offer more than this limited support for the teacher.

Some of the advantages that peer assessment brings for the instructor include: readily grade solutions in classes with a high number of students enrolled; quickly identify the levels of proficiency of students on a given topic [1] and discover learners that need remedial activities; monitor students' progress along the semester; measure students' reviewing abilities and critical thinking skills [14]; reduce potential teacher bias; focus reviewing effort on student work that really needs an expert opinion, while delegating the rest of the tasks to other students. Peer assessment proved particularly useful in MOOCs (Massive Open Online Courses), where thousands of learners are enrolled and the task of evaluating so many students is very complex and time-consuming [13]. On the other hand, peer assessment can also have drawbacks, such as student bias, validity and reliability issues, which the instructor needs to address.

The current landscape of peer assessment systems is diverse, including platforms such as: CrowdGrader [1], Peer Grader [5], WebPA [7], CaptainTeach [8], SocialX [10], CritViz [11], Mechanical TA [14]. Many of these systems lack advanced features for instructor support such as: configuration of the different parameters involved in computing students' competence and reviewing scores; selection of the review allocation mechanism; an approach for identifying the submissions that need teacher's grading; charts and reports regarding review data statistics and learner models; automatic notifications when relevant actions occur in the platform; ability to handle review requests coming from the students and possibility to leave back-reviews.

In an attempt to address these limitations, we designed and implemented a comprehensive peer assessment system, called LearnEval. The platform aims to provide a broad range of features both for the students and for the teacher. The student module part of the system was already described in [2]; it includes an automatic score-computing mechanism, a reputation system for addressing reliability issues and an open learner model with suggestive graphical visualizations. In the current paper, we focus on the instructor support module, which provides extensive assignment management functionalities. We start with an overview of related platforms, outlining their teacher support features (section II). Next, we describe the LearnEval instructor module in terms of functionalities, architecture and implementation; we also present an initial experimental validation of the system in the context of a Multimedia Technologies in E-Learning project (section III). Finally, we end the paper with some conclusions and future research directions (section IV).

II. RELATED WORK

In what follows, we present an overview of the instructor support provided by existing peer assessment platforms, aiming to summarize current challenges and potential solutions.

Mechanical TA [14] is an automated peer assessment system that involves human teaching assistants as a means to guarantee high quality reviews. Students start in a "supervised" state and are advanced to an "independent" state when they prove that they are able to comprehend the review criteria and are capable of using them correctly. The teaching assistants (TAs) verify students that did not prove reviewing proficiency and spot check the rest; "spot checking" is the process in which the system automatically assigns a subset of solutions and their reviews to be analyzed by the TAs. In addition, the essays assigned a grade above a configurable threshold are automatically reviewed by the TAs to mitigate the risk of students assigning high grades to every peer. Teachers' grading duties are supported by a calibration mechanism that helps students to provide high quality reviews from the start. The TAs workload is also reduced by assigning students in the "supervised" state to review essays submitted by peers in the same state; this is also valid for the

students in the "independent" state. This limits the number of solutions assigned to supervised reviewers, which need to be evaluated by the TA.

SocialX [10] is an exercise sharing and assessment tool which integrates a reputation system that fosters motivation and increases the interaction between students. The instructor can analyze the reputation earned by students with respect to the selected topic or course and gain insights into the learning progress of the students. An important aspect represents the capability of the teacher to configure the weights of the metrics involved in computing the reputation scores: involvement, usefulness to others, competency on the topic, ability to judge others' solutions and critical thinking. The instructor can flag a solution as "good" and its author will earn reputation when others reuse his/her solution; conversely, the instructor can flag a solution as "bad" and the students will earn reputation by spotting errors and offering the correct solution.

ARISE [13] is a guidance system for peer review to improve the efficiency of large scale grading, especially for tasks that are complex. An engine automatically scaffolds the paper to be evaluated based on rules that are defined by the teacher such that important concepts and pointers of higher order thinking are drawn to the regard of the reviewer.

CritViz [11] offers support for the teacher to overcome the difficulties in using only objective criteria for assessment in classes with creative work. The system is especially designed to scale up for large classes, where it is difficult to arrange discussions and meetings for offering support and feedback to students. CritViz integrates basic functionalities for the instructor, such as creating assignments, adding questions with various response types, and selecting a critique assignment algorithm.

WebPA [7] is a peer assessment system that tries to solve the problem of assigning individual grades to the students in a team and can be used on any kind of group assignment regardless of the discipline. The instructor can configure the number and size of the teams, as well as the assessment workflow and criteria. The teacher must follow three steps: specify the assignment requirements, create the groups of students, and link the groups with the assignments to create assessments. The review data is confidential and only the instructor can view it and assign group marks based on it. These group marks are used by the system to automatically assign grades to each individual in a group. WebPA generates comprehensive reports detailing the scores and allows the teacher to query the system for anomalous marks, but also to configure the algorithm that is applied for computing the individual scores. The top advantages for the instructor as reported in the paper are: reduces the grading burden and the number of complaints from the learners, makes the interaction within the student group outside the contact periods more transparent, increases the confidence in the peer assessment process, provides automatically generated individual scores and reduces the number of calculations. One of the limitations of the system is that the teacher must cope with the modifications in the structure of a group, such as withdrawal of a student from a course. Therefore, instructors must verify the solutions manually in the groups where learners do not provide the evaluations.

SPARK [4] is a web-based template for self and peer assessment aiming to increase the fairness of team evaluations and improve students' learning in group tasks. It automates the logistics of data gathering and calculation, reducing the workload for teachers. Another benefit for instructors mentioned in the paper is the lower number of team issues requiring teacher's intervention.

Emarking [12] is a collaborative grading platform that supports summative and formative assessment in higher education and includes distinctive features for printing management, scanning support, markers training, peer reviewing and on-screen-marking. As with the rest of the systems, the main aim of the platform is to reduce the time necessary to provide high quality feedback. The printing module allows the instructor to securely upload an exam as a PDF file and the server will send commands for printing it and optionally store it in a sealed envelope. The scanning module allows students' answers to be uploaded into the platform; for identification of the answers two QR codes are added to each page. Reusable comments can be defined by the teacher based on frequent errors that are expected to arise. For ease of use, the assessment rubric can be imported from Excel.

Several other peer assessment systems have been proposed in the literature, such as: [1], [6], [8]. However, these are focused mainly on student functionalities and do not offer support for the instructor; they are generally based on predefined peer assessment scenarios, which are run without teacher's intervention; the instructor does not have the opportunity to configure the workflow, review allocation or grade computing mechanisms.

Overall, starting from the literature analysis as well as our own practical experience with peer assessment [9], we summarize a list of support features highly desirable for the instructor: configurable assessment process by means of a settings module, various mechanisms for allocating the submissions to reviewers, configurable scoring system for addressing different competences of the students, automatic assignment of grades based on peer review data to reduce the grading burden, algorithms for identifying the solutions that need teacher intervention, advanced reporting system with suggestive graphical visualizations and comprehensive notifications module. Our aim is to integrate these functionalities in our general-purpose peer assessment platform, called LearnEval. The student module part of the system has already been presented in [2]; in the next section we describe the functionalities, implementation and pilot study of the teacher module prototype.

III. LEARNEVAL INSTRUCTOR MODULE PROTOTYPE

LearnEval provides a comprehensive assignment management module for the instructor. The teacher can create a course and enroll students to it (out of those registered in the platform). He/she can then design different peer assessment scenarios and create various assignments associated to the course. The system offers a detailed monitoring of student activity, including various scores and statistics; an open learner model is computed by LearnEval and made available to the instructor (consisting of the *involvement*, *competence* and *reviewing abilities* of the student). The peer assessment settings are highly parameterized, so the instructor can configure the scenario and scores according to the specificities of the course.

A. Main Functionalities

The Course page offers the instructor access to the following features: creation and editing of assignments, visualization and review of student solutions, various assignment settings, detailed statistics, scores and notifications, as presented next.

1) Create assignment

Upon creating an assignment the teacher must specify a name for it, provide the content in the form of a URL or uploaded file, submission deadline, review deadline, number of reviewers required for each submission and review criteria.

2) View assignments and solutions

This module allows the instructor to visualize the list of assignments and download, edit or delete an assignment. The teacher may also navigate to a page where they can view the solutions submitted by the students along with the grades received (as shown in Fig. 1); each grade is accompanied by a colored dot indicating the degree of confidence computed by the system based on the reviewing skills of the evaluator students, as described in [2]. The instructor may also visualize the student reviews for a given submission or submit his/her own review.

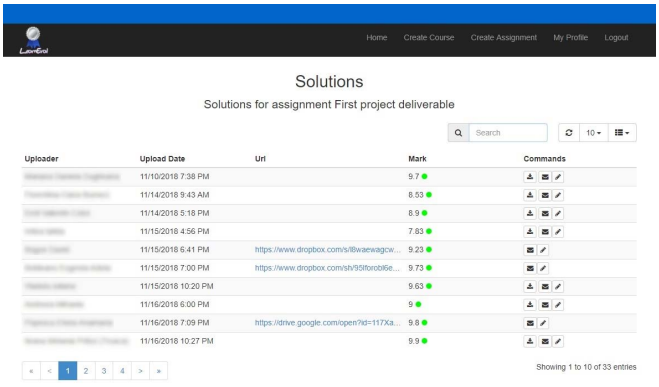

Fig. 1. LearnEval - List of solutions submitted by the students

3) Settings

This module represents the engine of the system, allowing the instructor to configure various parameters of the peer assessment process. The parameters are grouped in several panels (sections), based on the metric they have an influence on, such as solution score, involvement, competence, reviewing and overall scores of the student (as shown in Fig. 2).

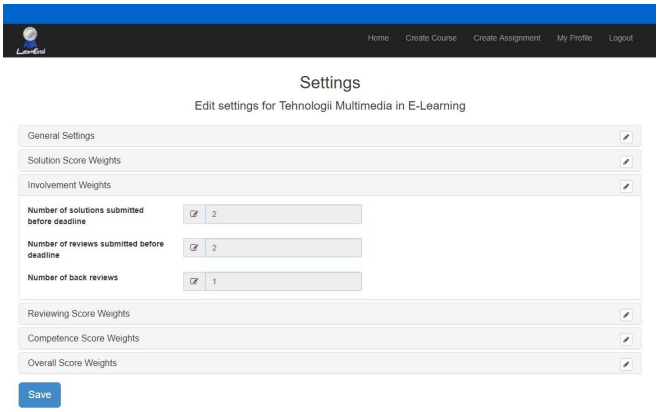

Fig. 2. LearnEval - Settings module

The *General Settings* panel allows the instructor to configure the anonymity of reviewers, anonymity of solution authors and the review allocation mechanism. Although in order to reduce student bias it is generally recommended to hide the identity of the reviewers and authors, we decided to let this decision in teachers' hands, so that they can apply different instructional scenarios depending on the context and the type of class. When it comes to allocation of the solutions to reviewers, the system currently supports several mechanisms: automatically based on splitting students in categories according to their reviewing skills, manually by the teacher, manually by the student and randomly by the system. Thus LearnEval platform is very versatile, allowing the instructor to choose between different pedagogical designs depending on the context of the course, the size of the class, the reviewing experience and competence of the students or the grading load of the teacher.

The *Solution Score Weights* panel allows the instructor to configure the weights of the metrics involved in computing the grade for a solution: teacher mark, average mark of evaluations done by students with high reviewing skills (HRS), average mark of evaluations done by students with medium reviewing skills (MRS) and average mark of evaluations done by students with low reviewing skills (LRS). By default, if the teacher mark is present, then it is given a high weight, counting for 70% of the final mark; students' marks are weighted according to their reviewing skills (3, 2 and 1, respectively).

The *Involvement Score Weights* panel allows the instructor to configure the weights of the metrics used for computing the student involvement score: the number of solutions submitted by the student before the deadline (denoted as *ns*), the number of reviews submitted by the student before the deadline (*nr*) and the number of back-reviews done (*nb*). Normalized values (on a 1 to 10 range) are provided for each of the three metrics. Each metric is assigned a weight by the teacher (denoted *wi1*, *wi2* and *wi3* respectively), and the involvement score is computed as follows:

$$InvolvementScore = \frac{wi1 * ns + wi2 * nr + wi3 * nb}{wi1 + wi2 + wi3}$$

Upon creating a course, the default values for the weights *wi1*, *wi2* and *wi3* are 2, 2 and 1 respectively, as we consider submitting solutions and reviews more important than providing back reviews.

The *Reviewing Score Weights* panel allows the instructor to configure the weights of the metrics involved in computing the student reviewing score: the average score of the back-reviews received by the student from peers (*pbra*) and from the teacher (*tbra*) and the agreement of the student's reviews with the final mark assigned to the reviewed solutions (*afm*). The *afm* metric is computed as follows:

$$afm = 10 - \frac{\sum_{i=1}^{n} |ms_i - m_i|}{n},$$

where: n is the number of solutions reviewed by the student, ms_i is the mark assigned by the student to solution i and m_i is the final mark assigned by the system to solution i. Again, each metric is assigned a weight by the teacher (denoted *wr1*, *wr2* and *wr3* respectively), and the reviewing score is computed as follows:

$$ReviewingScore = \frac{wr1 * pbra + wr2 * tbra + wr3 * afm}{wr1 + wr2 + wr3}$$

The default values for the weights *wr1*, *wr2* and *wr3* are 1, 1, and 2 respectively, as we consider the closeness to the actual mark of the solution to be the most important.

The *Competence Score Weights* panel allows the instructor to configure the weights of the metrics involved in computing the competence score: the average of the marks received from the teacher (*atm*), the average mark of the submitted solutions (*ams*) and an overall average mark (*amo*), which is calculated taking into account also the missed assignments (i.e., solutions not submitted, graded with 1). Each metric is assigned a weight by the instructor (denoted *wc1*, *wc2* and *wc3* respectively), and the competence score is computed as follows:

$$CompetenceScore = \frac{wc1 * atm + wc2 * ams + wc3 * amo}{wc1 + wc2 + wc3}$$

The default values for the weights *wc1*, *wc2* and *wc3* are 2, 1, and 1 respectively, as we consider the marks received from the teacher more relevant than the marks assigned by the peers.

Finally, the *Overall Score Weights* panel allows the instructor to configure the weights of the previous three scores (denoted *wo1*, *wo2* and *wo3* respectively) for computing the overall score of each student:

$$OverallScore = (wo1 * InvolvementScore + wo2 * CompetenceScore + wo3 * ReviewingScore)/(wo1 + wo2 + wo3)$$

The default values for the weights *wo1*, *wo2* and *wo3* are 1, 1 and 1 respectively, as we consider that each score plays an equally important role.

Given that all the metrics for computing involvement score, reviewing score and competence score are in the range 1 to 10, all four scores have values between 1 and 10; this leads to a direct mapping to the 1 to 10 grading scale, which is intuitive for both the teacher and the students.

4) Statistics

This module provides suggestive graphical visualizations of the course data and it is divided in two sections: *General Course Statistics* and *Student Statistics*.

The *General Course Statistics* section provides relevant information regarding the class as a group. In the following, we will provide a description of each of the visual components that convey information to the teacher in order to get better insights into the class activity. *Number of students per review category* tab displays in a pie chart the distribution of students according to their review skills: HRS, MRS and LRS. *Solutions submitted before deadline* component displays in a column chart the number of solutions submitted before deadline compared with the number of solutions not submitted before deadline for each assignment; the comparison allows the instructor to grasp the percentage and number of students that provide in time solutions. Similarly, *Reviews submitted before deadline* tab displays in a column chart the number of reviews submitted before deadline compared with the number of reviews not submitted before deadline; the comparison provides the teacher an insight into the review activity of the students.

Condensed information regarding the assignment grades is also provided. Thus, the *Average mark of solutions* component depicts in a line chart the evolution of the average grade for the solutions submitted for each assignment; this helps the instructor to spot assignments that were too simple or too complex. The *Passing marks* tab displays in a column chart the number of submissions that received a mark greater than 5, compared to the number of submissions that received a mark lower than 5, for each of the assignments; this provides the teacher with a quick overview of the passing / failing status of the class. Finally, *Solutions marks* tab summarizes in a column chart the marks received by each of the enrolled students for the selected assignment (as shown in Fig. 3).

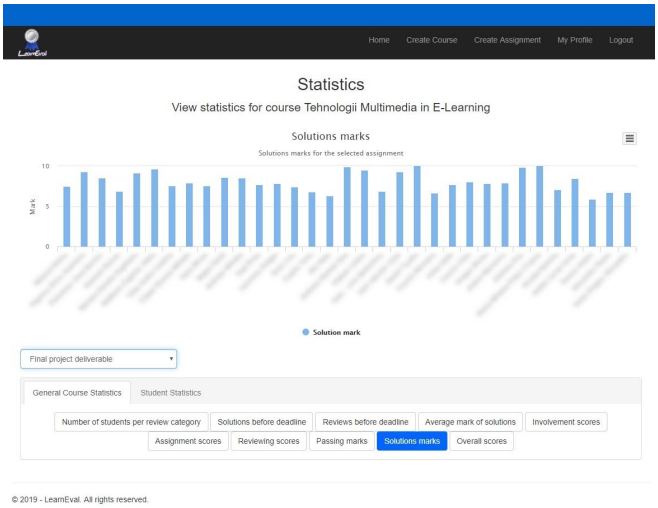

Fig. 3. LearnEval - Solution marks chart

Learner model information is also made available to the instructor, in concise graphical format. Thus, the *Involvement scores* tab shows in a column chart the involvement score for each student enrolled in the course; hence, the teacher can see at a glance the students that are not actively participating in the course and take remedial actions. *Assignment scores* component depicts in a column chart the average solution score submitted compared to the average solution score overall for each student; this allows the instructor to easily view the students that submit high or low quality work and draw conclusions regarding the overall knowledge level of the class. Similarly, the *Reviewing scores* tab displays in a column chart the reviewing score for each of the enrolled students, providing an overview of the class assessment skills. Finally, the general scores of the students are summarized in a column chart, in the *Overall score* component.

The *Student Statistics* section provides relevant information regarding a selected student enrolled in the course. The charts are similar with the ones that can be visualized by the student, as described in [2]; they illustrate the evolution of the grades throughout the semester, comparisons between the different grades received for the same solution and criteria breakdown, number of reviews or back-reviews received for each submission.

5) Scores

This module gives the instructor access to each learner model, providing a detailed breakdown of the involvement, reviewing and competence scores; average class values are also included for each metric, so that the teacher can position the student within the cohort. An excerpt from a student

score page is presented in Fig. 4. An alternative graphical representation of the learner model is also provided, based on visualization components such as progress bars, gauges and medals.

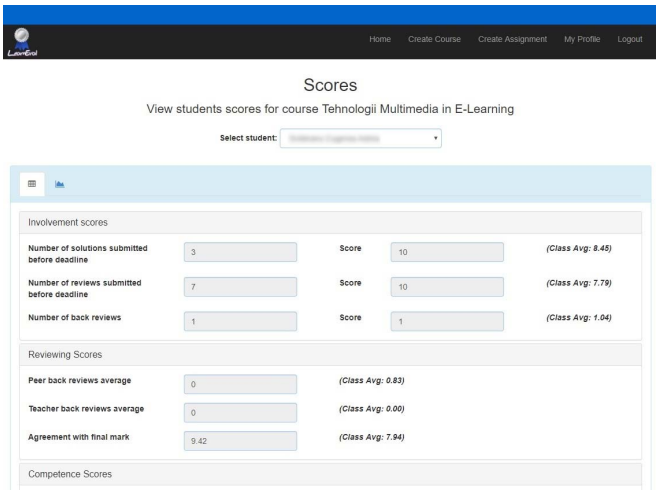

Fig. 4. LearnEval - Scores module

In addition, a *Reviewing Statistics* section is included, offering descriptive statistics regarding the marks given by the selected student to his/her peers' work: *range*, *standard deviation, interquartile range*, *mean*, *median*, *kurtosis*, *skewness*, as well as *consensus with teacher*; *consensus with peers* and *consensus with final mark*. These provide the instructor with an overview of the reviewing profile of each student.

6) Notifications

This module allows the teacher to view two general types of notifications: *system notifications* and *requests for expert opinion* from the students; an illustrative screenshot of the module is included in Fig. 5.

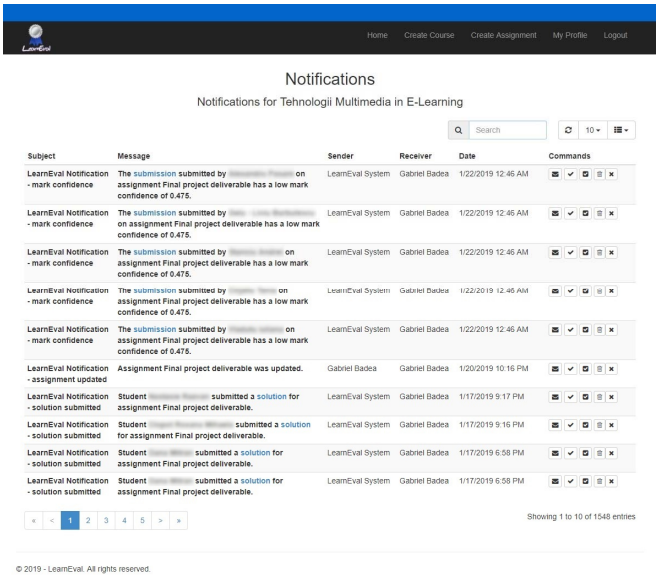

Fig. 5. LearnEval - Notifications module

System notifications are messages automatically generated by the system when a relevant action for a student or instructor occurs in the platform, such as: a reviewer receives a back review, the submission/review deadline for an assignment is due in X hours, a student receives a review from a peer/teacher for a solution submitted by him/her, a student submitted a solution to an assignment, a student is enrolled to a course, a final mark is assigned to a submission, a submission has a low mark confidence or manual allocation (by students) of solutions to reviewers has started. The instructor has access to all the notifications, including those aimed at the students; however, in order to prevent overload, the module provides a robust management mechanism, allowing for various filter, search and sort options, as well as marking and deleting notifications in batches.

The second type of notifications, *requests for expert opinion*, are appeals made by the students to the teacher to evaluate their submissions. This handles the cases when students are not satisfied with the reviews received from their peers, due to different causes such as: inconsistent grading, low quality of the review, inappropriate content or lack of feedback motivating the grades received. By default, each student is initially allowed only three such requests for expert opinion, to ensure that only justified appeals are made and reduce the burden on the instructor. Each request must be accompanied by a rationale, providing students' explanation for the appeal. If the teacher finds the request well justified, the student is granted an additional expert request token.

The module also offers the instructor the possibility to directly send an email to the recipient of a notification. The message fields are pre-filled with the notification text, which can subsequently be edited by the teacher. Thus, the subject of the email is replaced with the notification title (e.g. "*LearnEval Notification – assignment uploaded*"), while the body is replaced with the notification message (e.g., "*Assignment Homework 3 was updated.")*. The message is in HTML format for a proper display in the browser; an important enhancement is the automatic integration of links to specific LearnEval pages in the pre-filled messages. In the previous example, when pressing on "Homework 3", the student is redirected to the page where they can view the assignment details. Similar links are provided for other types of messages, such as: main course page when a student is enrolled to a course, specific solution page when a solution is uploaded by a student or when a final mark is assigned to a solution, reviews page when a student receives a back-review, assignment page when a submission deadline is approaching, submissions to review page when a review deadline is approaching etc.

B. Architecture and Implementation

LearnEval web application was developed in C# using ASP.NET MVC 5 framework and an SQL Server database. To achieve a flexible, extensible and reusable platform, the application is partitioned in multiple layers such as:

- *Presentation Layer*, which contains the views and controllers of the application. The views are .cshtml files that combine C# code with HTML.
- *Business Layer*, which contains the main logic of the application such as the services. A service was developed for each application entity.
- *Data Access Layer*, which contains the database context, database migrations, application entities and repositories. A repository was developed for each application entity. The Repository pattern was used to create an abstraction layer between the business layer and data access layer and achieve a higher separation of concerns.

JavaScript and libraries such as jQuery and Knockout were used for the client side of the application. The user interface is friendly and responsive, the platform being easily accessible from mobile devices.

C. Pilot Study

LearnEval prototype was used in a real classroom setting, in a pilot study involving 41 undergraduate students in Multimedia Systems Engineering, from the University of Craiova. The study took place in the first semester of 2018-2019 academic year, in the context of a Multimedia Technologies in E-Learning project. The platform successfully supported the teacher to manage the three assignments created along the semester: define the tasks; provide the submission and review deadlines; specify the review criteria and number of reviewers per solution; configure various course settings such as the weights of the teacher and peer marks, anonymity of reviewers and solution authors or the review allocation mechanism. The notification module proved very valuable, especially for drawing attention to the solutions having low confidence marks. Furthermore, the statistics and score modules allowed the instructor to easily monitor students' activity, identify learners who were not involved in the project and get an overview of student skills. Detailed results of the pilot study will be reported in a forthcoming paper.

IV. Conclusion

The paper presented the instructor module of the LearnEval peer assessment platform. The module integrates a comprehensive set of support features for the teacher: a highly configurable course settings component in order to customize the peer assessment scenario; an advanced reporting module with suggestive graphical visualizations in order to gain insights into the student progress and activity; a detailed scores component in order to visualize the learner model and take remedial actions if necessary; a performant notifications module in order to keep track of all the important actions occurring in the system. The platform was used in a pilot study, proving very helpful for the instructor of a Multimedia Technologies in E-Learning project.

Further experimental studies of LearnEval in real classroom settings are planned to be conducted. The instructor module could be extended with a mechanism for detecting rogue reviews; this would allow the teacher to recognize cases when identical feedback is provided to different review criteria or inappropriate feedback is given. Furthermore, including a calibration module would help students to offer high quality reviews from the beginning and reduce the workload of the instructor.

V. References

[1] L. D. Alfaro and M. Shavlovsky, "CrowdGrader: Crowdsourcing the Evaluation of Homework Assignments", Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE 2014), pp. 415–420, 2014.

[2] G. Badea and E. Popescu, "A Web-Based Platform for Peer Assessment in Technology Enhanced Learning: Student Module Prototype", Proceedings of the 19th International Conference on Advanced Learning Technologies (ICALT 2019), pp. 372-374, 2019.

[3] M. Boubouka and K. A. Papanikolaou, "Alternative Assessment Methods in Technology Enhanced Project-Based Learning", International Journal of Learning Technology, vol. 8(3), pp. 263-296, 2013.

[4] M. Freeman and J. McKenzie, "SPARK, A Confidential Web-Based Template for Self and Peer Assessment of Student Teamwork: Benefits of Evaluating across Different Subjects", British Journal of Educational Technology, vol. 33(5), pp. 551-569, 2002.

[5] E. Gehringer, "Electronic Peer Review and Peer Grading in Computer Science Courses", Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001), pp. 139 - 143, 2001.

[6] C.E. Kulkarni, R. Socher, M. S. Bernstein, and S. R. Klemmer, "Scaling Short-Answer Grading by Combining Peer Assessment with Algorithmic Scoring", Proceedings of the 1st ACM Conference on Learning @ Scale (L@S 2014), pp. 99–108, 2014.

[7] S. Loddington, K. Pond, N. Wilkinson, and P. Willmot, "A Case Study of the Development of WebPA: An Online Peer-Moderated Marking Tool", British Journal of Educational Technology, vol. 40(2), pp. 329–341, 2009.

[8] J.G. Politz, D. Patterson, S. Krishnamurthi, and K. Fisler, "CaptainTeach: Multi-Stage, In-Flow Peer Review for Programming Assignments", Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITICSE'14), pp. 267-272, 2014.

[9] E. Popescu and L.M. Petrosanu, "Integrating a Peer Evaluation Module in a Social Learning Platform", Innovations in Smart Learning, Lecture Notes in Educational Technology, Springer, pp. 141-150, 2016.

[10] M. Temperini and A. Sterbini, "Learning from Peers: Motivating Students through Reputation Systems", Proceedings International Symposium on Applications and the Internet (SAINT 2008), pp. 305-308, 2008.

[11] D. Tinapple, L. Olson, and J. Sadauskas, "CritViz: Web-Based Software Supporting Peer Critique in Large Creative Classrooms", Bulletin of the IEEE Technical Committee on Learning Technology, vol. 15(1), pp. 29-35, 2013.

[12] J. Villalon, "Emarking: A Collaborative Platform to Support Feedback in Higher Education Assessment", Foundations and Trends in Smart Learning, Lecture Notes in Educational Technology, Springer, pp. 113-118, 2019.

[13] A. Vista, E. Care, and P. Griffin, "A New Approach towards Marking Large-Scale Complex Assessments: Developing a Distributed Marking System that Uses an Automatically Scaffolding and Rubric-Targeted Interface for Guided Peer-Review", Assessing Writing, vol. 24, pp. 1–15, 2015.

[14] J.R. Wright, C. Thornton, and K. Leyton-Brown, "Mechanical TA: Partially Automated High-Stakes Peer Grading", Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15), pp. 96–101, 2015.