CONCEPTUAL ARCHITECTURE OF AN AUTOMATED SYSTEM FOR ASSESSING STUDENTS' ASSIGNMENTS

Stefani Paunova, Vladimir Valkanov, Angel Georgiev

Abstract. *Working in the field of education is invariably accompanied by homework, exams and their checks. Nowadays, we are benefited by numerous systems giving the ability to dynamically prepare check and homework assignments, assigning them to a group of people and providing the possibility of transmission and completion from different points of the world. One of these commonly used systems is Google Classroom. Among others of this kind, they allow the compilation of a grading scale and automatic checking of predefined answers. Their disadvantage is the lack of checking of assignments consisting of free text. With this paper, we propose a conceptual architectural solution to the posed problem. The idea of the application is to analyze free-response tasks and to evaluate them without the direct intervention of a human hand.*

Key words: Student Assessments, Automation, Conceptual Architectures, SQL, homework tasks.

Introduction

The fast progress of digital technology has changed education by creating web-based learning systems that provide major benefits in accessibility, scalability, and efficiency. Google Classroom, Moodle, Blackboard, and Schoology are now essential tools for overseeing coursework, providing educational material, and enabling communication between educators and students. These platforms gather course materials, assignments, and feedback in one place, improving accessibility and organization in various learning settings.

Although they have advantages, current systems frequently do not fully meet the need for more advanced features in assessment, specifically in automating the assessment of free-text responses like essays or open-ended questions. For instance, Google Classroom provides simple grading features but does not have customizable grading scales, automated assessment abil-

ities, or advanced performance analytics [2, 6]. Teachers frequently have to rely on manual assessment, which can be time-consuming and is not easily adaptable, particularly for classes with a high number of students.

This paper suggests a theoretical framework for a automated system to improve the grading capabilities of Google Classroom. The main objective is to create a solution that is both scalable and efficient while easily integrating with current features. The system being suggested aims to offer automated grading for open-ended responses, customizable grading choices, and advanced analytics, ultimately enhancing the educational process for both students and educators. This research seeks to fill the gaps in current educational technology platforms, providing a model that combines innovation and practicality to benefit the larger field.

Existing web-based systems for students' course management

Web-based learning management systems (LMS) have revolutionized the educational landscape by offering platforms that centralize content delivery, assignment tracking, and communication between educators and students. Popular systems like Google Classroom, Moodle, Blackboard, and Schoology provide essential tools for managing coursework and engaging learners in both synchronous and asynchronous environments [1, 11]. Google Classroom, for instance, has gained widespread adoption due to its seamless integration with other Google Workspace tools and its user-friendly interface. It allows teachers to distribute assignments, collect student work, and provide feedback digitally. However, it is limited in terms of advanced assessment functionalities, such as automated grading and detailed performance analytics [2, 6].

Moodle, another prominent LMS, is an open-source platform known for its flexibility and modular design. It supports a wide range of plugins and extensions that allow educators to customize the learning environment to suit their pedagogical needs. While highly versatile, the platform's complexity can pose challenges for educators without technical expertise [1, 11]. Blackboard and Schoology cater primarily to higher education and K–12 institutions, respectively. They focus on providing comprehensive solutions that include course material delivery, performance tracking, and student analytics. However, like Google Classroom, they often require additional integrations or third-party tools to perform advanced functionalities such as automated essay grading or plagiarism detection [4, 6, 11].

Despite their differences, all these platforms share common limitations. Current systems excel in content management and communication but are less effective in providing robust assessment tools. The reliance on manual grading, particularly for free-text responses and essays, highlights the need for systems capable of automating the evaluation process while maintaining accuracy and fairness [3, 5]. This section establishes the context for exploring enhancements to these platforms, particularly in extending their capabilities for automated assessment, as discussed in this paper.

Methodology

The proposed approach focuses on extending the grading capabilities of Google Classroom by integrating an automated system that evaluates student submissions. This system leverages the Google Classroom API and webhooks to detect when an assignment is submitted. Once a submission is detected, the system processes the assignment, evaluates its correctness, and updates the grade in Google Classroom automatically. This enhances the platform's grading functionality by introducing automated feedback, scalable evaluation, and support for complex assignments such as SQL queries [7].

The system architecture comprises the following components:

1. **Google Classroom Integration:** Utilizes webhooks to trigger actions when assignments are submitted and uses the Google Classroom API for fetching submissions and updating grades.
2. **Custom Python API:** Processes notifications from webhooks and validates the syntax of submitted queries and executes the validated queries against a predefined database and compares the results with expected outputs.
3. **Database Component:** Dynamically creates and manages database schemas and tables based on assignment requirements and executes queries and validates student results.
4. **Feedback Mechanism:** Generates detailed feedback for each query and sends the final evaluation results to Google Classroom.

The architecture ensures seamless integration, real-time processing, and accurate grading while minimizing manual intervention from teachers.

The tools and technologies used for the proof of concept are: Python,

SQL, Google API, “sqlparse” and etc.

This methodology ensures that the system effectively bridges the gap in Google Classroom’s current grading functionality, providing a scalable, efficient, and user-friendly solution for educators and students alike. On Figure 1 it is depicted the location of the configuration guides for integrating webhooks/push notifications in applications outside the Google ecosystem.

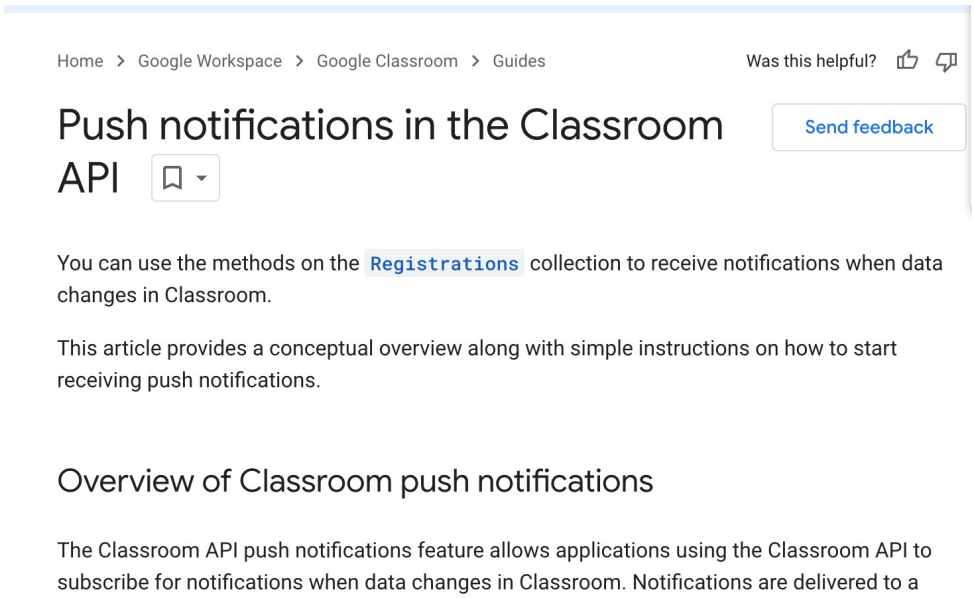

Figure 1. Push notification in Google Classroom

Architecture Model

The proposed architecture aims to automate the assessment of student assignments by integrating Google Classroom with a custom backend system. The design ensures seamless communication between Google Classroom, the backend API, and a SQL database for query execution and result validation. An example model is depicted in Figure 2.

The system components are the following:

1. **Data Input**
   - **Source:** Student assignments submitted via Google Classroom.
   - **Input Mechanism:** Webhooks provided by Google Classroom notify the custom backend API when a submission occurs. These notifications include assignment metadata such as

the student ID and the submitted file.

2. **Preprocessing:** the backend API reads and parses the submitted queries (e.g., SQL scripts) for syntax and structure. Queries are validated for correctness using tools such as sqlparse to ensure they conform to SQL standards. Valid queries are sent to a dynamically created database schema in SQL for execution.
3. **Grading Engine:** the SQL database executes the queries submitted by the student. The outputs of the executed queries are compared to pre-defined correct results stored in the system. Points are awarded based on the accuracy of the query results and compliance with the expected query structure.
4. **Feedback Mechanism:** detailed feedback, including errors (if any), is generated for each query to guide the student. Once all queries are evaluated, the final score is sent back to Google Classroom via its API, automatically updating the student's grade.

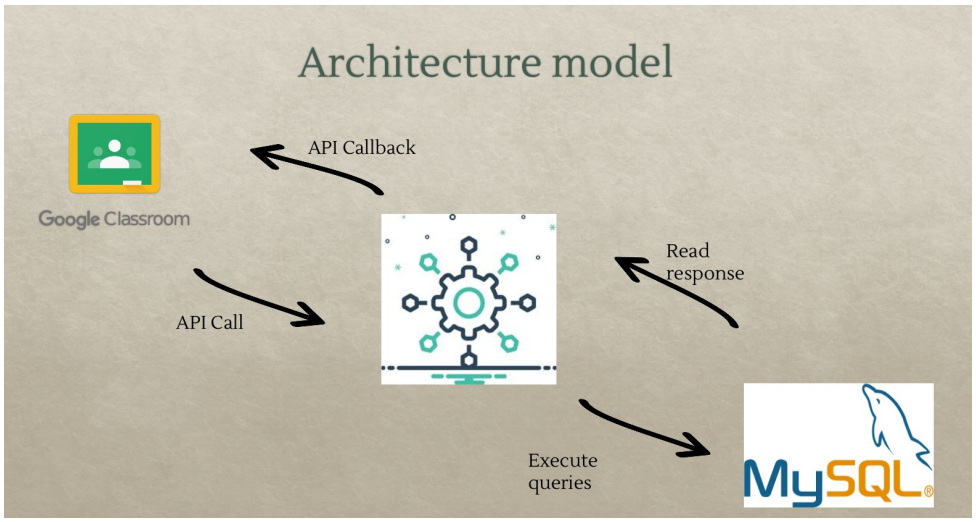

Figure 2. Example architecture model

The workflow is as it follows: Classroom sends notification to API, than the API parses then assignment, after that the valid queries are executed against a dynamically created SQL database. The next step is generating feedback and finally updating the grades in Classroom.

There are several challenges in developing integrations with Google Classroom like: API Limitations, webhook/notification configuration, data security and privacy and database scalability. This system is designed

to be scalable, efficient, and user-friendly, addressing key gaps in existing Google Classroom functionalities while offering automated grading tailored for SQL-based assignments.

Conclusion and Future Work

This paper introduced a theoretical framework to enhance Google Classroom capabilities through the automation of grading student assignments. A custom Python backend, the Google Classroom API, and a SQL database are utilized by the system to simplify assignment assessment. The combination of automatic query execution and grading addresses a notable deficiency in Google Classroom, providing teachers with a useful tool for managing extensive class sizes and challenging assignment formats. The suggested system offers numerous important advantages: quicker response, enhanced scalability, improved objectivity. In order to enhance the system and expand its functionalities, the following advancements are anticipated:

- Assistance for Complex SQL Functions
- Enhance grading features to assess stored procedures, functions, and packages.
- Implementing PL/SQL alongside T-SQL expands the range of database systems where it can be utilized.
- Incorporation of Artificial Intelligence

Utilize machine learning algorithms for managing advanced grading responsibilities, such as assessing procedural code and identifying instances of plagiarism [3, 11].

Utilize AI-powered analysis to offer personalized feedback that adjusts to the unique needs of each student [8].

- In-depth analysis of learning data: Create sophisticated dashboards for educators, providing information on trends in student performance and pinpointing areas that need enhancement [8, 11].
- Improved integration of API: Improve how the custom API and Google Classroom interact for easier bulk grade updates and better feedback integration.
- Enhanced Protection of Data and Privacy: Improve encryption protocols to protect student data while being processed and transmitted, ensuring adherence to data privacy laws like GDPR [9].

- Scalability of System and Optimization of Resources: Implement strategies like database caching and load balancing for efficiently managing concurrent submissions from large groups of students.

This research outlines a conceptual architecture for enhancing Google Classroom's grading functionality by introducing an automated system for evaluating student assignments. By leveraging the Google Classroom API, a Python-based backend, and a SQL database, the proposed solution addresses key challenges such as manual grading inefficiencies, scalability for large class sizes, and lack of advanced analytics. This work lays the foundation for further innovation in automated assessment systems, bridging gaps in existing platforms and fostering a more adaptive and effective learning process.

Acknowledgments

The paper is funded by the Project FP23-FMI-002 of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

References

[1] J. Spector, M. Merrill, J. Elen, M. Bishop, *Handbook of Research on Educational Communications and Technology*, Springer, New York, 2014, ISBN: 978-1-4614-3185-5.

[2] T. Kingsley, Challenges in integrating Google Classroom with third-party tools, *Journal of Online Learning*, Vol. 15, No. 3, 2021.

[3] R. Luckin, W. Holmes, M. Griffiths, L. Forcier, *Intelligence Unleashed: An Argument for AI in Education*, Pearson, London, 2016, ISBN: 978-1-2921-4195-5.

[4] B. Nguyen, T. Ha Thi, H. Chen, ANALYZING GIT LOG IN AN CODE-QUALITY AWARE AUTOMATED PROGRAMMING ASSESSMENT SYSTEM: A CASE STUDY, *Tra Vinh University Journal of Science*, Vol. 13, No. 3, 2023, ISSN: 2815-6072, E-ISSN: 2815-6099, DOI: 10.35382/tvujs.13.3.2023.2430.

[5] M. Khan, A. Ahmad, M. Humayoun, Building an Effective Automated Assessment System for C/C++ Introductory Programming Courses in ODL Environment, *arXiv*, 2022, 222–234, ISBN: 978-988-8238-13-2, https://doi.org/10.48550/arXiv.2205.11915.

[6] Google Developers, Classroom API Documentation, Google Cloud, 2024. Available at: https://developers.google.com/classroom,

(Last visited: 04.12.2024).

[7] R. Binns, Fairness in Machine Learning: Lessons from Political Philosophy, *Proc. of Machine Learning Research 81:149-159, 2018 Conference on Fairness, Accountability, and Transparency*, 2018, https://doi.org/10.48550/arXiv.1712.03586.

[8] G. Siemens, P. Long, Penetrating the Fog: Analytics in Learning and Education, *EDUCAUSE Review*, 5, 2011, 30–32, DOI: https://doi.org/10.17471/2499-4324/195.

[9] D. Solove, Privacy self-management and the consent dilemma, *Harvard Law Review*, Vol. 126, No. 7, 2013, https://harvardlawreview.org/print/vol-126/introduction-privacy-self-management-and-the-consent-dilemma/, (Last visited: 04.12.2024).

[10] I. Roll, R. Wylie, Evolution and Revolution in Artificial Intelligence in Education, *International Journal of Artificial Intelligence in Education*, Vol. 26, 2016, 582–599, DOI: https://doi.org/10.1007/s40593-016-0110-3.

[11] N. Malik, A. Bilal, M. Ilyas, S. Razzaq, F. Maqbool, Q. Abbas, Plagiarism Detection Using Natural Language Processing Techniques, *Technical Journal*, University of Engineering and Technology (UET) Taxila, Pakistan, 26 (1), 2021, ISSN: 1813-1786, E-ISSN: 2313-7770.

Stefani Paunova[1], Vladimir Valkanov[2], Angel Georgiev[3],
[1,2,3] Paisii Hilendarski University of Plovdiv,
Faculty of Mathematics and Informatics,
236 Bulgaria Blvd., 4027 Plovdiv, Bulgaria
Corresponding author: angel.georgiev@uni-plovdiv.bg