

QUESTIFY: An Intelligent Application for Creating, Submitting and Assessing Open-Ended Questions*

Ana-Ioana POPESCU, Livia OLARU, Bianca-Gabriela DOBRE, Beatrice-Iuliana UTA and
Maria-Iuliana DASCALU

National University of Science and Technology POLITEHNICA Bucharest,
Faculty of Engineering in Foreign Languages, Bucharest, Romania, ana_ioana.popescu@stud.fils.upb.ro

Correspondence should be addressed to: Maria-Iuliana DASCALU, maria.dascalu@upb.ro

*Presented at the 47th IBIMA International Conference, 29-30 June 2026, Madrid, Spain

Abstract

The assessment of open-ended questions remains a challenging and time-consuming task in higher education, requiring significant teacher effort and often leading to inconsistencies in feedback and evaluation. This paper presents Questify, an intelligent educational platform designed to support the complete lifecycle of open-ended question assessment, from question creation and test organization to answer evaluation, feedback generation, and progress monitoring.

The proposed solution integrates JavaFX, PostgreSQL, Spring Boot REST services, and artificial intelligence techniques to assist teachers in generating keywords, estimating question difficulty, creating evaluation rubrics, and producing personalized pedagogical feedback. Unlike many existing educational platforms that focus primarily on grading or test management, Questify combines AI-assisted assessment support, learning analytics, automated communication, and student progress tracking within a unified environment.

The application follows a modular architecture and supports multiple user roles, including students, teachers, and administrators. Experimental validation demonstrates the feasibility of integrating AI-assisted educational services into a desktop-based assessment platform while maintaining the teacher as the final decision-maker in the evaluation process.

The results indicate that Questify can reduce repetitive assessment activities, improve feedback consistency, and provide students with more transparent and personalized learning support. The proposed approach illustrates how artificial intelligence can be effectively integrated into educational software to enhance open-ended question assessment without replacing human judgment.

Keywords: artificial intelligence, automatic assessment, intelligent feedback, open-ended questions.

Introduction

The assessment of open-ended answers represents an important challenge in modern educational systems. Unlike multiple-choice questionnaires, open-ended answers cannot be assessed solely by comparison with a predefined answer. They require an analysis of content, coherence, argumentation, and the student's level of understanding. This process is time-consuming for teachers and may sometimes introduce a degree of subjectivity into the correction process, as noted by Zhai et al (2021), Ramesh and Sanampudi (2022), and Süzen et al (2020).

In the context of the digital transformation of education, digital tools can contribute to improving the creation, organization, and evaluation of pedagogical content. Artificial intelligence technologies also offer new possibilities for assisting teachers with repetitive tasks, such as generating keywords, preparing evaluation rubrics, estimating question difficulty, or writing personalized feedback, as discussed by Russell and Norvig (2021), Zhai et al (2021), and Al Faraby and Romadhony (2024). However, many educational platforms remain focused on traditional tests and do not always provide a complete environment for managing open-ended questions.

In this context, Questify was developed as an intelligent desktop application built with JavaFX and connected to a PostgreSQL database, as documented by PostgreSQL Global Development Group (2025) and Oracle (2025). The application centralizes the entire pedagogical process: creating open-ended questions, organizing tests, publishing content, submitting student answers, evaluating responses, and consulting results. It provides separate spaces for teachers, students, and administrators, each with rights and functionalities adapted to their role.

Questify also integrates several advanced features. Teachers can automatically generate keywords, estimate the difficulty of a question, obtain an evaluation rubric, and produce AI-assisted feedback. Students can consult their tests, submit their answers, import text from different types of files, consult their grades, track their progress, and send feedback by email. The application also uses a REST API developed with Spring Boot, as described by Walls (2018) and Fielding (2000) to support feedback delivery through Gmail SMTP.

The main objective of this project is to propose a complete, modern, and scalable solution for the assessment of open-ended questions. Questify does not replace the role of the teacher, but assists teachers in correction, feedback structuring, and performance analysis. The system aims to improve the efficiency of the evaluation process while offering students clearer, more personalized, and more useful support for their learning, as shown by Zhai et al (2021) and Al Faraby and Romadhony (2024).

Open-Ended Question Assessment

The automatic assessment of open-ended questions has become a significant research topic in Artificial Intelligence in Education (AIED). Unlike multiple-choice assessments, open-ended responses require semantic understanding, reasoning evaluation, contextual interpretation, and analysis of argument quality. Consequently, Zhai et al (2021) note that researchers have developed numerous approaches to support teachers in the assessment process.

Early assessment systems relied mainly on keyword matching, rule-based evaluation, and predefined answer templates. These methods offered fast processing and simple implementation but were limited in their ability to recognize paraphrases, alternative formulations, and deeper semantic relationships between concepts, as Ramesh and Sanampudi (2022) point out.

Advances in Natural Language Processing (NLP) introduced more sophisticated techniques such as text mining, semantic similarity analysis, and machine learning methods for evaluating textual responses. As Süzen et al (2020) showed, these approaches improved the assessment of short-answer questions by considering contextual relationships between words rather than exact keyword matches.

More recently, machine learning and deep learning techniques have significantly improved automatic assessment performance. Research on Automated Essay Scoring (AES) by Ramesh and Sanampudi (2022) demonstrates how intelligent systems can evaluate essays based on linguistic, semantic, and structural features, reducing assessment time while maintaining acceptable accuracy levels. Transformer-based models and Large

Language Models (LLMs) have further expanded the possibilities of educational assessment by enabling semantic understanding, content generation, and automated feedback generation, as shown by Al Faraby and Romadhony (2024).

Modern educational platforms integrate AI-assisted functionalities such as rubric generation, automated grading suggestions, plagiarism detection, learning analytics, and personalized feedback generation, as discussed by Zhai et al (2021) and Al Faraby and Romadhony (2024). Recent studies have also emphasized explainable AI and teacher-in-the-loop approaches, where artificial intelligence supports the assessment process while preserving human supervision and final decision-making, noted also by Zhai et al (2021).

Despite these advances, fully automated evaluation remains challenging. Russell and Norvig (2021) and Ramesh and Sanampudi (2022) point out that open-ended responses often require contextual understanding, domain-specific knowledge, and pedagogical judgment that cannot always be reliably captured by AI systems. Furthermore, Al Faraby and Romadhony (2024), emphasize that concerns regarding transparency, consistency, fairness, and explainability continue to represent active research directions in educational AI.

Although many existing systems focus primarily on grading or feedback generation, fewer solutions provide a complete environment that supports the entire assessment lifecycle, including question creation, rubric generation, answer submission, feedback generation, statistical analysis, and progress monitoring. Questify addresses this need by integrating these functionalities into a unified platform while maintaining the teacher as the final decision-maker in the evaluation process.

Table 1: Comparison between Questify and existing educational assessment platforms

Feature	Moodle (2025)	Google Forms	Gradescope (2025)	Questify
Open-ended question support	Yes	Basic support	Yes	Yes
AI-generated keywords	No	No	No	Yes
AI-generated evaluation rubric	No	No	Partial	Yes
AI-assisted feedback generation	No	No	Partial	Yes
Question difficulty estimation	No	No	No	Yes
Student progress analytics	Yes	Basic support	Yes	Yes
PDF feedback export	Basic support	No	No	Yes
Feedback by email	No	No	No	Yes
Complaint management	No	No	No	Yes

As shown in Table 1, Questify differs from existing educational assessment platforms by combining open-ended question management, AI-assisted rubric generation, feedback generation, difficulty estimation, progress analytics, and communication features in a single integrated environment.

Main Features of Questify

Questify provides a complete set of functionalities for managing the process of creating, submitting, and assessing open-ended questions. A demo of the application is available here: <https://youtu.be/sRhe4Inys20>. The application is structured around three main categories of users: teachers, students, and administrators. Each user type has access to specific features according to their role, which, as Fowler (2002) and Gamma et al (1994) suggest, ensures a clear separation of responsibilities within the system.

Teacher space allows teachers to create open-ended questions, define expected keywords, estimate the difficulty of a question, generate evaluation rubrics, organize questions into tests, publish tests, consult submitted answers, assign grades, and provide feedback. Teachers can also use artificial intelligence features in order to reduce repetitive work and improve the consistency of the assessment process, as shown by Russell and Norvig (2021), Zhai et al (2021), and Al Faraby and Romadhony (2024). These AI-assisted features do not replace human decision-making, but support teachers in structuring the evaluation process more efficiently.

The student space allows students to consult published tests, submit answers, import text from different types of files, consult grades and comments, follow their progress, export feedback as PDF, and send feedback by email. As Zhai et al (2021) and Al Faraby and Romadhony (2024) note, students can also request personalized advice generated by artificial intelligence in order to better understand their mistakes and improve future answers. In addition, they can submit complaints when they identify a problem related to a grade or feedback. This functionality supports communication between students and teachers and makes the assessment process more transparent.

The administrator space is dedicated to user management. The administrator can validate teacher accounts, manage users, and remove accounts when necessary. This role contributes to the security and organization of the platform by controlling access to the system, according to Fowler (2002) and Gamma et al. (1994).

The functional analysis of Questify is therefore based on three main actors: the student, the teacher, and the administrator. The student consults published tests, submits answers, consults grades, and can request improvement advice. The teacher creates open-ended questions, organizes tests, evaluates answers, provides feedback, and analyzes statistics. The administrator manages users and validates teacher accounts.

This distribution of responsibilities makes it possible to guarantee a clear separation of access rights. Each user has access only to the features corresponding to their role. For example, as Fowler (2002) and Gamma et al (1994) would suggest, a student cannot evaluate an answer, while a teacher cannot freely delete all user accounts, as seen in Table 2.

Table 2: Main Components of Questify and Their Responsibilities

Component	Category	Purpose
Student Module	User Interface	Answer submission, grade consultation, progress monitoring
Teacher Module	User Interface	Question creation, assessment, feedback generation
Administration Module	User Management	Account validation and user administration
PostgreSQL Database	Data Layer	Persistent storage of educational data
Spring Boot REST API	Integration Layer	Email-based feedback delivery
AI Assistance Module	Intelligence Layer	Keyword generation, rubric generation, difficulty estimation, pedagogical feedback

Figure 1 presents the main use cases of Questify and shows the interaction between the three categories of users and the system. The diagram highlights the most important functionalities available for each actor, such as question creation, test publication, answer submission, feedback generation, complaint management, and user validation.

The general workflow of Questify begins when the teacher creates open-ended questions and organizes them into tests through the JavaFX interface, as described by Oracle (2025). After a test is published, students can access it and submit their answers either directly as text or by importing documents. The application supports several input formats, which makes the submission process more flexible and closer to real educational situations.

Once the answers are submitted, they are stored in the PostgreSQL database, as documented by PostgreSQL Global Development Group (2025), and can be consulted by the teacher. The teacher may use the AI-assisted functionalities to generate keywords, evaluation criteria, and pedagogical feedback, as shown by Russell and Norvig (2021), Zhai et al (2021), and Al Faraby and Romadhony (2024). However, the final assessment remains under the teacher's control, since the artificial intelligence module is used as an assistance tool and not as a replacement for human judgment.

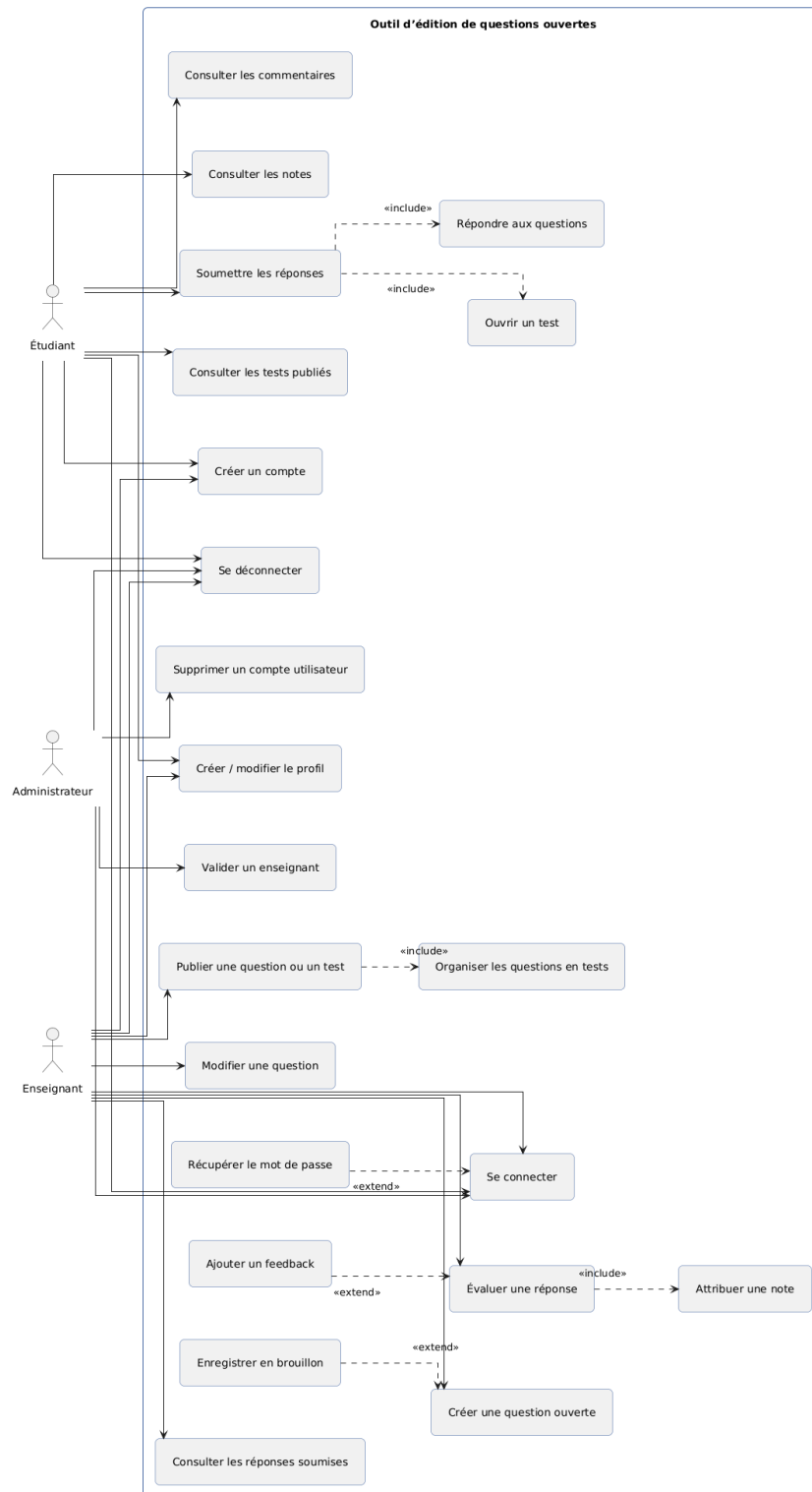


Fig 1. Use Case Diagram

After the evaluation is completed, students can consult their grades and feedback in the application. They can also export their feedback as a PDF document or send it by email through the Spring Boot REST API (Walls (2018); Fielding (2000)). This workflow demonstrates the integration of several components of the system: the JavaFX interface, the PostgreSQL database, the AI module, and the REST service used for external communication.

Figure 2 illustrates the sequence of operations involved in the creation of an open-ended question. The teacher enters the question data in the interface, the controller validates the information, the service layer applies the business logic, and the DAO layer saves the question in the PostgreSQL database. If the teacher requests AI assistance, the system can also generate keywords, estimate the difficulty of the question, or prepare an evaluation rubric.

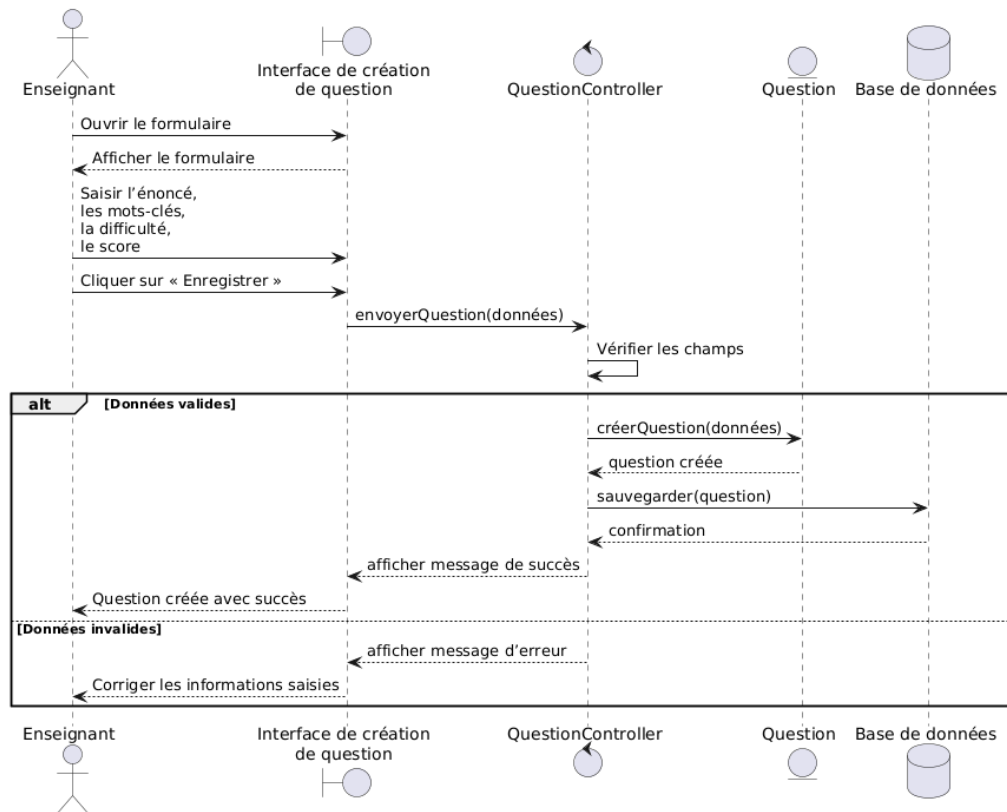


Fig 2. Sequence Diagram of the Open-Ended Question Creation Process

Questify also includes statistical features for both teachers and students. For teachers, the statistical dashboard can display information such as the total number of tests, the number of created questions, the number of submitted answers, the number of evaluated answers, the average grade, the number of generated feedback, and the number of complaints. These indicators help teachers better understand the activity of their students and identify possible difficulties in the learning process.

These data can also be represented through visual elements, such as charts showing the number of answers per test or the distribution of grades. Such visualizations help teachers interpret student activity more quickly and make pedagogical decisions based on the recorded data.

For students, the progress page displays indicators such as the number of evaluated answers, the overall average, the best grade obtained, and the evolution of grades over time. This functionality encourages students to monitor their own progress and identify the areas in which they need to improve.

The main purpose of these statistical features is to transform educational data into actionable information. In this way, Questify does not only support the submission and assessment of open-ended answers but also provides a clearer view of the learning process for both teachers and students.

General Architecture of the Application

Questify is organized according to a modular architecture, in order to clearly separate the user interface, business logic, data access, and external services. The main application is developed in JavaFX and provides a

modern graphical interface intended for three categories of users: teachers, students, and administrators, as described by Oracle (2025).

The modular organization of Questify also facilitates the future evolution of the application. Each component has a well-defined role, which makes it possible to add new features without modifying the entire system. For example, the modules related to artificial intelligence, PDF export, or email sending can be improved independently of the main graphical interface. As Fowler (2002) and Gamma et al (1994) suggest, this approach reduces coupling between components and improves the overall maintainability of the project.

Moreover, the use of JavaFX combined with PostgreSQL and Spring Boot demonstrates the possibility of integrating several modern technologies into a single coherent educational platform. This architecture makes Questify closer to professional applications used in academic and industrial environments.

The data access layer uses the DAO model in order to communicate with a PostgreSQL database, as described in the documentation provided by PostgreSQL Global Development Group (2025). This database stores users, tests, questions, answers, grades, feedback, and complaints. The use of prepared statements helps secure exchanges with the database and reduce the risk of SQL injections.

The application is also connected to a REST service developed with Spring Boot, following the principles described by Walls (2018) and Fielding (2000). This external service enables the automatic sending of feedback by email. Thus, Questify is not limited to a local application, but adopts architecture closer to modern professional systems, in which several software components communicate with one another.

The general architecture can be summarized as follows: the JavaFX interface enables interaction with the user, the controllers process actions, the services apply the functional logic, the DAOs ensure data persistence, and the REST API manages external communication with Gmail SMTP and Fowler (2002) and Gamma et al (1994) outline similar architectural principles.

Questify's architectural model is based on a three-layer organization: the presentation layer, the business logic layer, and the data layer. This choice was defined from the intermediate project documentation in order to clearly separate responsibilities and facilitate the future evolution of the application, as Fielding (2000) and Fowler (2002) discuss.

The presentation layer corresponds to the JavaFX interfaces and FXML files. It allows teachers, students, and administrators to interact with the system. The business logic layer includes the controllers and services, which are responsible for data validation, the management of business rules, calls to artificial intelligence modules, and the coordination of operations. Finally, the data layer is based on DAO classes and the PostgreSQL database, where accounts, questions, tests, answers, grades, feedback, and complaints are stored.

In order to improve the organization and maintainability of the code, several principles inspired by design patterns were used in Questify. The DAO pattern, Data Access Object, makes it possible to separate database access operations from the business logic. Thus, each important entity of the system has a dedicated class for managing SQL queries, such as UserDAO, TestDAO, or EvaluationDAO.

The Singleton principle is used for certain shared connections and services, in order to avoid the unnecessary creation of multiple instances and to centralize certain resources. In addition, several services play a role similar to the Facade pattern, because they simplify access to complex functionalities, especially for AI feedback generation, PDF export, or email sending through the REST API.

Fowler (2002), Gamma et al (1994), Bloch (2018), and Sierra, Bates and Gee (2022) also describe these principles as ways to improve the readability of the project, reduce coupling between components, and facilitate future extensions of the application.

This architecture makes it possible to avoid mixing the graphical interface, functional processing, and database access. It also makes the application more maintainable, because a modification of the interface does not necessarily require a modification of the business logic or of the database structure.

Technologies Used

Questify integrates several technologies in order to provide a complete educational platform for creating, submitting, assessing, and managing open-ended questions. The main application is developed in JavaFX, which allows the creation of a modern desktop graphical interface for teachers, students, and administrators (Oracle, 2025). The JavaFX interface is connected to controllers and services that manage user actions, data validation, file processing, database communication, and external service integration.

Another important technical feature of Questify is the possibility for students to submit their answers as files. The application can process several formats, including text files, Word documents, PDF files, and images. For text files, the content is read directly. For Word documents, the application extracts the paragraphs, while for PDF files, the text is retrieved using specialized libraries. In the case of images or scanned PDF files, Questify uses an OCR mechanism in order to identify and extract the text present in the image. This feature is particularly useful when students have written their answers by hand or when they already have a scanned document. The extracted text is then saved in the PostgreSQL database and can be used by the teacher or by the artificial intelligence module for assessment and feedback generation.

PostgreSQL Global Development Group (2025) describes this database, which stores user accounts, roles, tests, questions, submitted answers, evaluations, grades, feedback, complaints, and activity history. It stores user accounts, roles, tests, questions, submitted answers, evaluations, grades, feedback, complaints, and activity history. The main tables are connected in order to represent the complete pedagogical workflow. For example, an answer is linked to a student, a test, and a question, while an evaluation is linked to a specific answer. This structure allows students to consult their grades and feedback after the assessment process is completed. The database also supports the generation of statistics, such as the number of created tests, submitted answers, evaluated responses, average grades, and complaints.

For database communication, Questify uses the DAO pattern, which separates data access operations from the business logic of the application (Fowler, 2002). Each important entity is managed through dedicated DAO classes, such as users, tests, questions, answers, evaluations, and complaints. This organization improves code readability and maintainability. Moreover, database operations use prepared statements in order to reduce the risk of SQL injection by separating user input from SQL commands. Each insertion, update, deletion, or selection operation is therefore handled in a more secure and structured way.

Questify also integrates a REST API developed with Spring Boot in order to enable the sending of feedback by email (Walls, 2018). This functionality illustrates the communication between a JavaFX desktop application and an external service. The workflow is simple: the student selects a feedback from the “My grades and comments” page and clicks the button for sending feedback by email. The JavaFX application builds an HTTP request containing the feedback information in JSON format, including the recipient, subject, and message content. This request is then sent to the local REST API.

The Spring Boot service receives the request, constructs an email message, and sends it using Gmail SMTP. This separation is important because the JavaFX application does not directly manage email sending. Instead, it consumes an external REST service, which makes the architecture clearer, more modular, and easier to extend. This functionality demonstrates the practical use of REST API concepts, HTTP communication, JSON data exchange, and integration with an external service, also discussed by Fielding (2000).

Data security and validation are also important aspects of the application. Questify handles sensitive educational information, such as user accounts, student answers, grades, feedback, and complaints. For this reason, several validation and protection mechanisms were integrated into the system. User authentication is based on a login system with username and password, while user roles control access to the available functionalities. Students, teachers, and administrators therefore have different permissions within the application.

The application also includes form validation mechanisms. When creating a question, organizing a test, submitting an answer, or sending feedback, required fields are checked before the operation is executed. This validation reduces input errors and improves the consistency of the data stored in the system. The email sending functionality uses Gmail SMTP authentication and an App Password configuration, which avoids the direct use of the main Gmail account password and improves the security of the external integration, as explained by Walls (2018) and Fielding (2000).

Overall, the technologies used in Questify contribute to the development of a modular, secure, and extensible educational application. JavaFX provides the desktop interface, PostgreSQL ensures data persistence, DAO classes organize database access, Spring Boot enables REST communication, Gmail SMTP supports feedback delivery, and OCR-based processing improves the flexibility of answer submission. Together, these technologies support the main objective of Questify: to provide a modern and intelligent environment for the assessment of open-ended questions.

Integration of Artificial Intelligence

Artificial intelligence is used in Questify as a pedagogical assistance tool integrated into several stages of the open-ended question assessment process. The AI-assisted functionalities are implemented using OpenAI GPT-4o through API integration. The model is used for keyword generation, question difficulty estimation, evaluation rubric creation, personalized feedback generation, and student improvement recommendations. Its purpose is not to replace the teacher, but to support the creation of questions, the preparation of evaluation criteria, the generation of feedback, and the personalization of student support.

The first use of artificial intelligence appears during the creation of open-ended questions. When a teacher creates a new question, the AI module can automatically generate a list of relevant keywords related to the topic. As Ramesh and Sanampudi (2022) and Süzen et al (2020) note, these keywords help define the expected content of a correct answer and can guide the teacher in formulating clearer evaluation expectations. For example, for a question related to JavaFX, the system may suggest keywords such as “JavaFX”, “FXML”, “controller”, “graphical interface”, “event handling” or “MVC”. The teacher can then review, keep, modify or remove the proposed keywords before saving the question.

A second AI-assisted functionality is the estimation of question difficulty. The application can analyze the wording of a question, the expected concepts, and the level of reasoning required in order to provide an approximate difficulty level. This feature helps the teacher better organize questions inside tests and balance the complexity of an assessment. For example, a question asking students to define a concept may be considered easier than a question requiring comparison, argumentation, or application in a concrete programming context, as discussed by Zhai et al (2021) and Al Faraby and Romadhony (2024).

Questify also uses artificial intelligence for generating evaluation rubrics. Based on the question text, the expected answer, the keywords, and the maximum score defined by the teacher, the AI module can propose a structured evaluation grid. This grid may include criteria such as understanding of the subject, accuracy of information, logical organization, use of appropriate terminology, clarity of expression, and completeness of the answer. The score distribution must remain compatible with the maximum score chosen by the teacher. In this way, as shown by Zhai et al (2021) and Ramesh and Sanampudi (2022), the evaluation process becomes more structured and more transparent.

Another important use of artificial intelligence is the generation of pedagogical feedback. After an answer is evaluated, Questify can generate a structured feedback message for the student. This feedback usually contains the strong points of the answer, the elements that need improvement, and recommendations for future answers. The objective is to provide students not only with a grade, but also with an explanation that can support their learning process. This makes the feedback more useful, especially in the case of open-ended answers, where students need to understand why their response was incomplete or unclear, as explained by Zhai et al (2021), Ramesh and Sanampudi (2022), and Al Faraby and Romadhony (2024).

Artificial intelligence is also connected to the file submission functionality. When a student imports an answer from a text file, Word document, PDF file, or image, the extracted text can be used by the AI module for feedback generation or personalized advice. This makes the system more flexible because students are not limited to writing directly inside the application. The extracted answer can still be analyzed and used in the same pedagogical workflow.

For students, Questify can also generate personalized improvement advice. Based on their evaluated answers, grades, and feedback, the application can provide suggestions that help students understand their mistakes and improve future responses. These recommendations may concern the structure of the answer, the use of key concepts, the clarity of explanations, or the need to provide more complete arguments, as noted by Zhai et al (2021) and Al Faraby and Romadhony (2024).

However, the use of artificial intelligence in Questify remains controlled. The teacher keeps the final responsibility for the assessment. AI-generated keywords, rubrics, difficulty levels, and feedback messages are considered suggestions that can be accepted, modified, or rejected. This approach is important because open-ended answers require interpretation, context, and pedagogical judgment. Therefore, Questify follows a teacher-in-the-loop approach, where artificial intelligence supports the evaluation process without replacing human decision-making, as described by Zhai et al (2021), Ramesh and Sanampudi (2022), and Al Faraby and Romadhony (2024).

Overall, the integration of artificial intelligence improves the efficiency and consistency of the assessment process. It reduces repetitive tasks for teachers, supports the creation of more structured feedback, and offers students clearer explanations and personalized learning guidance. Through these functionalities, Questify demonstrates how AI can be used in a practical and controlled way in educational software dedicated to open-ended question assessment, as shown by Russell and Norvig (2021), Zhai et al (2021), and Al Faraby and Romadhony (2024).

Questify addresses this gap by integrating the full lifecycle of open-ended question assessment into a unified teacher-in-the-loop framework, where artificial intelligence supports key pedagogical tasks such as keyword generation, rubric construction, feedback preparation, and progress analysis, while the teacher remains responsible for interpreting student responses, validating the proposed outputs, and making the final assessment decision.

Questify follows a teacher-in-the-loop approach in which artificial intelligence acts as an assistance tool rather than an autonomous evaluator. AI-generated keywords, difficulty estimations, evaluation rubrics, and feedback suggestions are always subject to teacher review, validation, modification, or rejection before being used in the assessment process.

This design preserves pedagogical responsibility and human judgment while benefiting from the efficiency and consistency provided by artificial intelligence. By keeping the teacher as the final decision-maker, Questify combines AI-assisted support with educational accountability and transparency.

The integration of LLM-based assistance also introduces important methodological and ethical limitations, particularly regarding fairness, hallucination, and reproducibility, since model-generated feedback may reflect implicit biases, produce inaccurate or unsupported explanations, or generate variable outputs for similar student responses, thereby requiring systematic teacher validation and transparent supervision.

Prototype Validation and Results

The development of Questify made it possible to obtain a functional application integrating several modern technologies. The system manages a complete evaluation cycle: test creation, publication, answer submission, evaluation, feedback generation, result consultation, PDF export, and feedback sharing by email.

The application improves the teacher's work by automating certain repetitive tasks, such as keyword generation, preparation of an evaluation rubric, question difficulty estimation, and initial feedback generation. It also improves the student experience by offering more detailed explanations, progress tracking, personalized improvement advice, and the possibility of reporting a problem related to a grade or feedback.

The integration of the REST API represents an important result, because it shows that the application can communicate with an external service. This functionality brings the project closer to professional software architecture, where several components interact in a modular and extensible way.

The general statistics calculated from the data recorded in the Questify application are: total number of tests created, total number of questions created, total number of submitted answers, number of evaluated answers, overall average grade, number of feedbacks generated by AI, number of complaints sent, number of feedbacks sent by email. These indicators allow the teacher to have an overall view of pedagogical activity, including the number of tests, the number of submitted answers, the number of evaluated answers, the average grade, and the number of complaints. The main purpose of these statistics is to transform educational data into actionable information.

Figure 3 presents the distribution of functionalities by user type. The figure highlights that teachers and students have the largest number of functionalities, while administrators and external services support platform management and communication.

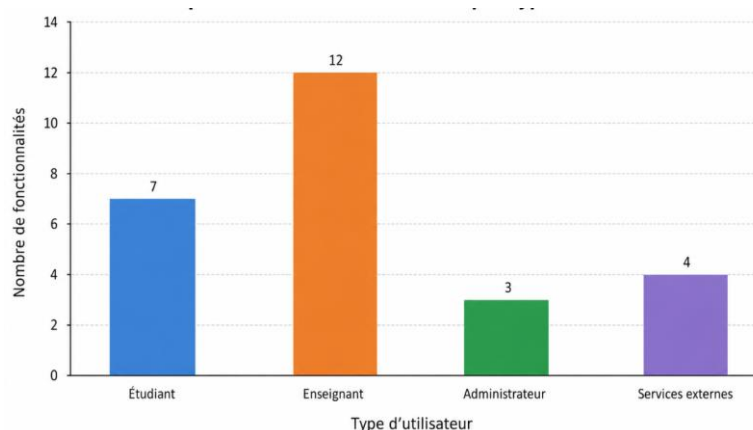


Fig 3. Distribution of Features by User Type

Questify does not aim to replace the teacher, but to reduce repetitive tasks and support a more structured, transparent, and efficient assessment workflow, as seen from Table 3.

Table 3: Comparison between manual assessment activities and Questify-assisted workflows

Assessment activity	Traditional/manual process	Questify-assisted process
Creating an evaluation rubric	Manually defined by the teacher	AI-assisted rubric generation
Identifying relevant concepts	Manual keyword selection	AI-generated keyword suggestions
Providing feedback	Individually written comments	AI-assisted feedback generation
Monitoring student progress	Manual consultation of grades	Automatic progress indicators and charts
Sending feedback	Manual email writing	Automated email delivery through REST API

The prototype was tested using a set of representative educational scenarios covering question creation, answer submission, evaluation, feedback generation, and progress monitoring. The validation scenarios covered question creation, AI-assisted keyword generation, rubric generation, answer submission, feedback generation, PDF export, and email delivery. All tested functionalities executed successfully and produced the expected results. The experiments confirmed the feasibility of integrating AI-assisted assessment services, learning analytics, and external communication services within a unified educational platform.

Application Interface and Functional

In order to better illustrate the functionalities of Questify, several screenshots of the application interface are included in this section. These visual elements show how the main features are implemented from the perspective of teachers and students and help demonstrate the practical use of the platform.

Figure 4 presents the teacher's space dedicated to the creation of open-ended questions. Through this page, the teacher can write the question, define the expected answer, set the maximum score, and use AI-assisted functionalities such as keyword generation, difficulty estimation, and evaluation rubric creation. This interface is important because it represents the starting point of the pedagogical workflow and shows how artificial intelligence supports the teacher during the preparation of assessment content.

Créer une question ouverte

Titre de la question

Énoncé de la question

Mots-clés

Niveau de difficulté

Score maximum

Enregistrer comme brouillon

Publier la question

Fig 4. Teacher Interface for Creating an Open-Ended Question with AI Assistance

Figure 5 illustrates the student space used for consulting grades, feedback, and progress. After an answer has been evaluated, the student can view the obtained grade, read the teacher's feedback, request personalized improvement advice, export the feedback as a PDF document, or send it by email. This interface highlights the student-centered dimension of Questify, since the application does not only provide a grade, but also offers explanations and recommendations that can support the learning process.

Mes notes et commentaires

Consultez vos réponses, vos notes et les feedbacks reçus.

Test	Question	Réponse	Note	Feedback
Les pays: Questions conçues p...	Les pays et leurs capitales	CHINA		En attente d'évaluation
Examen sur la programmation ...	Question générée par IA	Java est un langage de programmation orienté o... Avec Java, on peut créer : des applications desktop avec JavaFX ou Swing ; des applications web et des sites dynamiques ; des applications mobiles Android ; des serveurs et API backend ; des logiciels d'entreprise ; des jeux vidéo ; des systèmes embarqués et IoT ; des applications utilisant l'intelligence artificielle ...	10	Note proposée : 10 / 10 Points forts : - Expérience dans le domaine de Java et la programmation en général - Comprendre des aspects de programmation orienté objet - Connaissances sur les systèmes d'exploitation Java Points à améliorer : - Mise en évidence de l'intelligence artificielle avec l'exemple du traitement linguistique "Bonjour" - Explication des avantages d'utiliser la machine virtuelle Java (JVM) pour le développement web - Description des applications commerciales (logiciels d'entreprise, logiciels d'e-commerce) - Résumé de l'intelligence artificielle et les bases de données Feedback final : Bien que l'étudiant ait une bonne compréhension générale du langage de programmation Java, si Note : 9 / 10

Signaler un problème sur la note

Obtenir des conseils IA

Exporter feedback en PDF

Envoyer feedback par e-mail

Retour

Fig 5. Student Interface for Grades, Feedback, and Progress Tracking

Figure 6 presents the administrator space of Questify. This page is dedicated to platform management and gives access to functionalities such as teacher account validation and user account management. Through this interface, the administrator can review teacher accounts before validation and remove user accounts when necessary. This functionality contributes to the security, reliability, and proper organization of the platform by ensuring controlled access and centralized account administration.

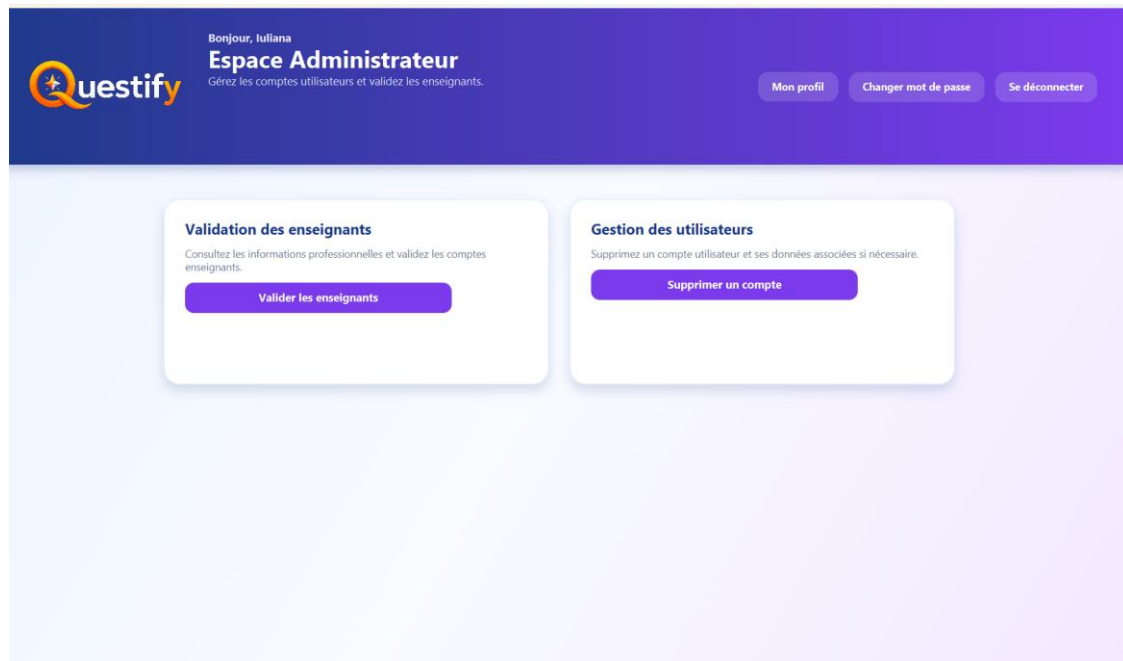


Fig 6. Administrator Interface for Teacher Validation and User Management

These screenshots demonstrate that Questify is not limited to a theoretical workflow but provides a complete and functional user interface. The application combines question creation, answer submission, AI-assisted feedback, progress tracking, and administrative control in a coherent educational environment.

Limitations and Perspectives

Even though Questify offers several advanced features, some limitations remain. The answers generated by artificial intelligence can sometimes be imprecise or require human validation. For this reason, the teacher remains responsible for the final decision.

Another limitation concerns dependence on external services. For example, sending emails requires the REST API to be running and the Gmail configuration to be correct. Similarly, some AI features depend on the model used and on the quality of the prompts.

In the future, Questify could be improved by integrating a more advanced system for detecting similarity between answers, adding a complete activity history, implementing a real-time notification system, and deploying the application in the cloud. A web version could also be developed in order to make the application accessible from a browser.

Conclusions

Questify is an intelligent educational application that combines JavaFX, PostgreSQL, artificial intelligence, and a REST API in order to improve the evaluation process for open-ended questions. The project responds to a real need: reducing correction time, structuring feedback, and offering students clearer monitoring of their performance.

The originality of the project lies in the integration of several complementary features: keyword generation, AI evaluation grids, personalized feedback, file submission, pedagogical statistics, PDF export, and sending feedback by email through a REST API.

The results obtained show that the application can effectively support the teacher's work while improving the student's learning experience. Questify does not replace human evaluation, but rather provides an intelligent, flexible, and scalable assistance environment, thus contributing to the broader landscape of AI-driven

educational applications that leverage artificial intelligence to support and enrich educational activities, according to Tarbă et al (2026).

References

- Bloch, J. (2018) Effective Java, 3rd ed., Addison-Wesley.
- Faraby, S. Al and Romadhony, A. (2024), 'Analysis of LLMs for educational question classification and generation,' *Computers and Education: Artificial Intelligence*, 7, Article 100298.
- Fielding, R. T. (2000), 'Architectural styles and the design of network-based software architectures,' Ph.D. dissertation, University of California, Irvine.
- Fowler, M. (2002) Patterns of Enterprise Application Architecture, Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.
- Gradescope (2025), 'Gradescope,' [Online], [Retrieved 23 June, 2026], <https://www.gradescope.com/>.
- Moodle (2025), 'Moodle,' [Online], [Retrieved 23 June, 2026], <https://moodle.org/>.
- Oracle (2025), 'JavaFX documentation,' [Online], [Retrieved 23 June, 2026], <https://docs.oracle.com/en/java/javase/components/javafx/>.
- PostgreSQL Global Development Group (2025), 'PostgreSQL documentation,' [Online], [Retrieved 23 June, 2026], <https://www.postgresql.org/docs/>.
- Ramesh, D. and Sanampudi, S. K. (2022), 'An automated essay scoring systems: a systematic literature review,' *Artificial Intelligence Review*, 55 (3), 2495-2527.
- Russell, S. and Norvig, P. (2021) Artificial Intelligence: A Modern Approach, 4th ed., Pearson.
- Sierra, K. and Bates, B. (2022) Head First Java, 3rd ed., O'Reilly Media.
- Süzen, N., Gorban, A. N., Levesley, J. and Mirkes, E. M. (2020), 'Automatic short answer grading and feedback using text mining methods,' *Procedia Computer Science*, 169, 726-743.
- Walls, C. (2018) Spring Boot in Action, Manning Publications.
- Zhai, J., Chu, X. and Chai, C. (2021), 'A review of artificial intelligence in education,' *IEEE Access*, 9, 1-15.
- Tarbă, N., Boiangiu, C.-A., Dumitrescu, E.-D., Gîrniță, A.-C., Popescu, M.-T., Tănase, M.-C., Voncilă, M.-L., & Goga, N. (2026). Artificial intelligence in education: Implications for the academic quality of future employees. *U.P.B. Scientific Bulletin, Series C*, 88(1), 5–20.