ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

Design and Implementation of an Integrated Training, Assessment, and Evaluation Intelligent Agent System Based on Tongyi Qwen - Max

Shuo Feng[1,*]

[1] College of Information and Intelligence, Hunan Agricultural University, Changsha, China, 410128

SUMMARY: *As the digital education strategy continues to advance, traditional practical training in higher education faces challenges such as heavy workloads for instructors and delayed feedback when addressing large-scale, personalized demands. This study designs and implements an integrated training, assessment, and evaluation intelligent agent system tailored to practical training scenarios in science and engineering disciplines at universities, aiming to drive the transformation of teaching models from experience-driven to intelligent through artificial intelligence technology. The system's core utilizes Alibaba Cloud's Tongyi Qwen - Max large language model, integrating retrieval-enhanced generation technology to deeply empower a local domain-specific knowledge base. It has established a comprehensive functional framework covering the instructor, student, and administrative sides. The instructor side supports knowledge-base-driven intelligent lesson preparation, diverse question generation, and automated grading; the student side provides real-time online Q&A, personalized assessments, and error analysis. Developed using a B/S architecture with SpringBoot, MyBatis-Plus, and Vue 3, the system ensures a modular design with high cohesion and low coupling, as well as robust data security. Application results demonstrate that this intelligent system effectively reduces the burden of repetitive teaching tasks for instructors, provides students with precise and immediate learning support, and establishes a closed-loop process encompassing practical training, assessment, and evaluation, thereby significantly enhancing the intelligence and overall efficiency of practical training instruction.*

KEYWORDS: *Large Language Models; Smart Education; Knowledge Base Management.*

1 Introduction

1.1 Software System Overview

This software is an intelligent teaching assistance platform for practical training teaching in colleges and universities[1-4]. The project is developed against the background of the in-depth advancement of the education digitalization strategy and the wide application of artificial intelligence technology. The traditional practical training teaching model faces many challenges in meeting large-scale and personalized teaching demands, such as heavy time consumption for teachers in lesson preparation and assessment design, delayed feedback on homework correction, and lack of targeted Q&A and practice for students[5, 6].

The core functions of the software cover three modules: teacher side, student side, and administrator side. The teacher side provides functions including intelligent lesson preparation

*per_2026@163.com
https://doi.org/10.65102/is20261278
Published: 10 June, 2026

based on the local knowledge base, diversified test question generation, automatic correction, and in-depth learning status analysis. The student side offers 24/7 online learning Q&A, personalized practice evaluation, and error analysis. The administrator side supports user management, teaching resource coordination, and large-screen visual monitoring[7].

The system is applied to course training, experimental teaching, and graduation project guidance for computer, software engineering, and related science and engineering majors in colleges and universities. The core value lies in the deep integration of open-source large model technology, turning artificial intelligence into an "intelligent teaching assistant" for teachers and a "full-time tutor" for students. It effectively reduces teachers' repetitive workload, improves teaching efficiency and quality, provides students with instant, accurate, and personalized learning support, and drives the transformation of practical training teaching from experience-driven to data-driven, intelligent, and adaptive[8, 9].

2 Software Design Constraints and Principles

2.1 Software Design Constraints

The software design is subject to the following constraints, and the design scheme must be optimized and balanced within these frameworks:

Hardware Constraints

Large model deployment environment: The system is built on Alibaba Cloud Bailian cloud-based large model services. The recommended server configuration includes a 4-core CPU, 8GB RAM, and 50GB storage space, significantly lowering the hardware requirements for server deployment.

Client environment: Users mainly access the system via web browsers, requiring the front end to be compatible with the latest versions of mainstream browsers (Chrome, Firefox, Edge), with no special requirements for users' local hardware.

Software Constraints

Development language and framework: Backend development is restricted to Java and the SpringBoot framework; front-end development is restricted to the Vue.js framework (Vue 3); the database is restricted to the MySQL relational database.

Large model selection: The system's core intelligent capabilities are implemented using the Qwen-Max model from Alibaba Cloud Bailian Tongyi Qwen series. Model integration is achieved via API calls, while also supporting enhanced RAG knowledge base retrieval functionality.

Operating system: The server side is recommended to be deployed on Linux distributions (e.g., Ubuntu 20.04/22.04 or CentOS 7/8) to ensure stability and compatibility.

Time and Cost Constraints

Time constraint: The overall development cycle is about 15 weeks, within which the whole process of design, development, testing, and deployment must be completed. The tight schedule requires reasonable design and clear module division to support parallel development.

Cost constraint: The initial project budget is limited. Software selection prioritizes open-source and free solutions, and hardware investment must be controlled within a reasonable range to avoid high commercial software licensing fees.

Compliance Constraints

Data privacy: The system must comply with the Personal Information Protection Law and other relevant laws and regulations, strictly protect the personal information and learning data of teachers and students, and encrypt the storage and transmission of sensitive data.

2.2 Software Design Principles

To ensure the quality and long-term maintainability of the software system, the following core principles are followed in this design:

Modularity with High Cohesion and Low Coupling

The system is divided into multiple independent modules according to business functions (e.g., user management, course management, lesson preparation design, Q&A center). Each module exhibits tightly integrated internal functionalities (high cohesion), while interacting with other modules through the Service and Controller layers to minimize dependencies (low coupling). This improves code readability, reusability, and testability.

Frontend-backend Separation

Strictly adhere to the front-end back-end separation architecture. The front-end focuses on user interface and interaction logic, while the back-end handles business logic, data processing, and RESTful API services. Data is exchanged between both parties in JSON format. This principle enables parallel development by front-end and back-end teams, enhances efficiency, and facilitates system scalability and maintenance[10].

Scalability

Extension points are fully reserved when designing database table structures, API interfaces, and class inheritance systems. For example, new question types can be easily added to the question_type field in the future, and the system_config table can dynamically adjust system behaviors without modifying code.

Maintainability

The design emphasizes code clarity, standardization, and complete documentation. A unified coding specification must be followed, and detailed annotations are added to key logic. This design document and subsequent API and deployment documents are important parts of system maintenance.

Security

Authentication and authorization: A JWT token mechanism is adopted for stateless user authentication to ensure secure access to API interfaces. Access permissions are strictly controlled for users with different roles (administrator, teacher, student).

Data security: The user password is encrypted and stored using the MD5 algorithm with salt.Key business data is backed up regularly. Common web attacks (e.g., SQL injection, XSS, CSRF) are prevented.

Business Integration Based on Open-source Large Models

The intelligent functions of the system are closely developed around the capabilities of large models. In lesson preparation design, the large model converts knowledge base content into structured teaching plans. In learning status analysis, the summarization and reasoning capabilities of the large model are used to analyze students' practice data and generate reports. The large model is deeply integrated into the teaching business flow rather than being a mere add-on function.

3 Software Design Scheme

3.1 Architecture Design

This software system employs a front-end/back-end separated architecture based on the B/S model, with modular decomposition to achieve high cohesion, low coupling, and excellent horizontal scalability[11-15]. The system is divided into four layers: front-end presentation layer, back-end service layer, data persistence layer, and AI service layer. The architecture diagram is as follows:

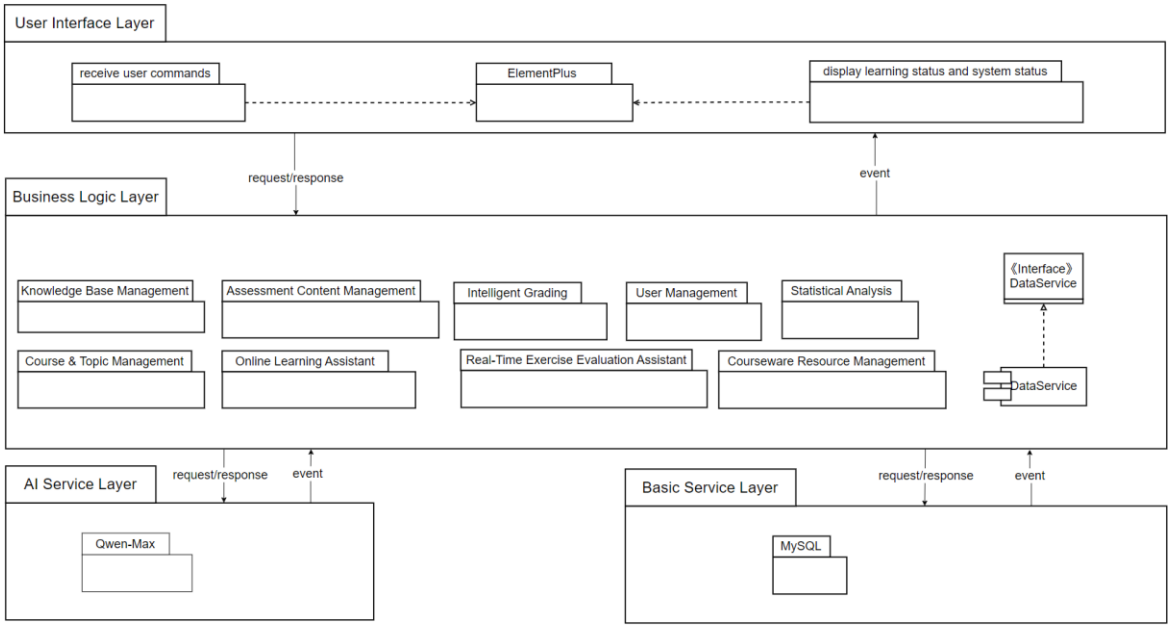

Figure 1: System Architecture Diagram - Package Map

System architecture diagram - package map is shown in Figure 1.

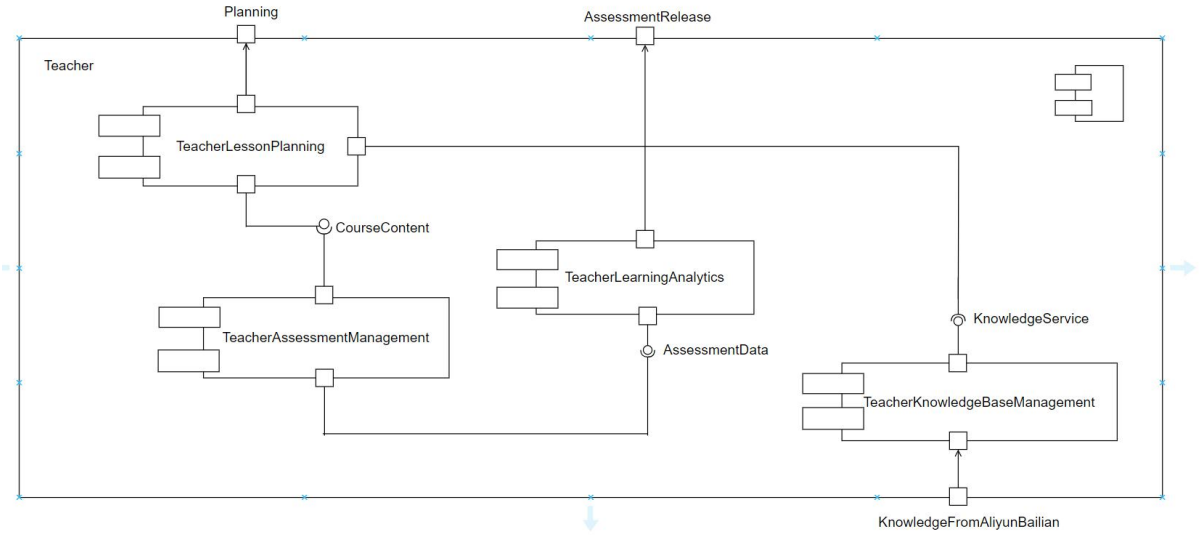

Figure 2: System Architecture Diagram - Teacher Module Component Diagram

System architecture diagram - teacher module component diagram is shown in Figure 2.

3.2 User Interface Design

User interface design follows the principles of clarity, intuitiveness, and efficiency, providing differentiated workbenches for three roles: teachers, students, and administrators. The overall style adopts the design language of modern educational technology products, dominated by blue and white tones to create a professional and focused learning atmosphere[16, 17]. The layout uses the classic "sandwich" structure of top navigation + sidebar menu + content area to

ensure concise operation paths and clear information hierarchy. The key interface design ideas and layout features are as follows:

Login page: The minimalist design focuses users' attention on the login process. At the center is a login card with username and password input fields and a login button. The background features abstract graphics related to education or technology.

Teacher-side lesson preparation design page: The core workspace for teachers, adopting a left-right split layout. The left panel is the control panel, including course selectors, knowledge base resource checkboxes, generation parameter settings, and core operation buttons. The right panel is the content editing and preview area with a rich-text editor, displaying teaching plan content generated by the large model for direct secondary editing and optimization by teachers.

Student-side Q&A center page: Designed to imitate mainstream instant messaging software to reduce students' cognitive burden. The left is the historical conversation list, the middle is the main conversation area clearly displaying students' questions and answers from the agent/teacher, and the right is the knowledge point recommendation area dynamically linking and displaying related course knowledge points and recommended learning resources.

Administrator-side large-screen overview page: Adopts a data visualization cockpit design with a dark background and high-bright colors for data cards and charts to highlight key indicators. The top displays KPI indicator cards, the middle shows trend analysis charts, and the bottom presents detailed data lists with filtering functions by subject, time, and other dimensions.

3.3 Use Case Design

Centering on three core actors (teacher, student, administrator), the system combs and designs core use cases covering the entire business process, fully including core functions such as intelligent lesson preparation, test question generation, online Q&A, practice evaluation, learning status analysis, user management, course management, and system management. Each use case defines the actor, use case description, precondition, postcondition, basic flow, and exception flow to ensure clear functional boundaries, complete business logic, and implementable interaction paths[18, 19].

As the core actor, teachers mainly complete teaching-related intelligent and management operations, including use cases such as intelligent lesson preparation generation, manual/batch test question import, intelligent test question generation, practice task release, correction result viewing, learning status analysis report viewing, and personal course and knowledge base management. For the intelligent lesson preparation generation use case, teachers must log in and enter the corresponding course, check the knowledge base, set generation parameters, and send a generation request. The system calls the large model to generate the teaching plan and returns editable content; in case of exceptions, a generation failure prompt is given and logs are recorded. The intelligent test question generation use case supports batch question generation by knowledge point, difficulty, and question type, automatically saving to the question bank with retry and content correction support for generation exceptions.

As the primary users, students' core use cases include online login, viewing courses and assignment tasks, submitting online answers, reviewing answer results and explanations, and posing questions to the intelligent agent. For online questioning, students must log in first and enter their questions. The system then searches the local knowledge base and utilizes the large model to generate precise answers while recording the conversation. If no results are found or the model fails to respond properly, the system provides a user-friendly prompt and guides students to rephrase their questions. After submitting assignments, the system automatically grades objective questions and evaluates subjective questions using the large model, delivering real-time scores, correctness feedback, and error analyses to ensure immediate feedback[20].

As the system operations and management role, administrators' core responsibilities include user management (adding, disabling, or modifying teacher and student information), knowledge base management, data statistics, and dashboard monitoring. After identity authentication, administrators can centrally manage all platform users, courses, and resources, supporting batch import and data export. All operations are fully traceable, with abnormal actions triggered by interception and logging to ensure system stability and data security.

All use cases follow the design rule of "precondition verification → core process execution → postcondition implementation → exception handling". Combined with use case diagrams, the association between actors and use cases is intuitively displayed, ensuring no missing system functions, no broken processes, and interactions matching real teaching and training scenarios.

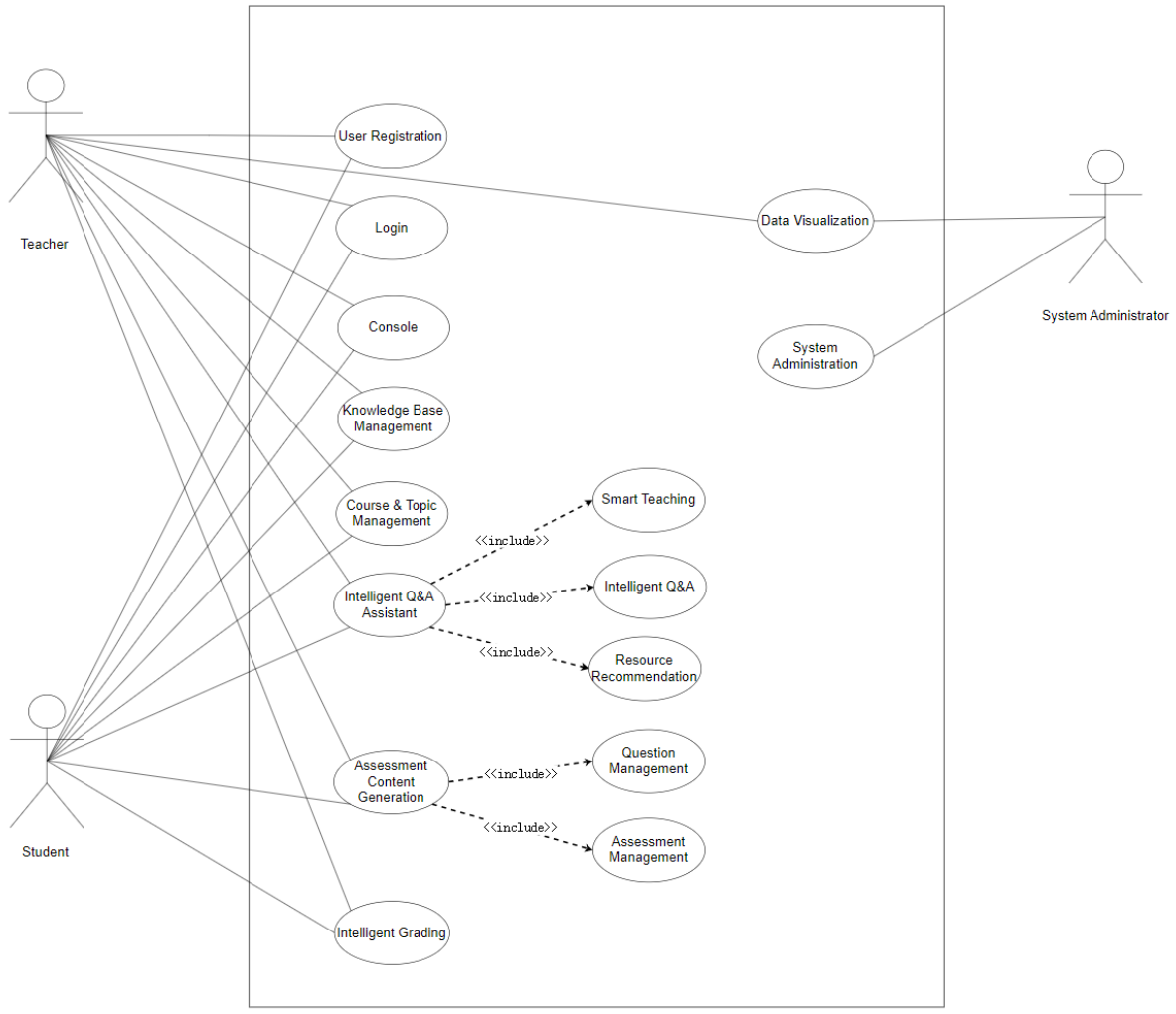

Figure 3: System Top-Level Use Case Diagram

System top-level use case diagram is shown in Figure 3.

3.4 Subsystem/Component Design

According to the architecture design, the back-end service layer can be split into the following core subsystems/components. All components collaborate through well-defined Java interfaces.

User Management Component: Responsible for registration, login authentication, information management, and permission control of all users. Interfaces: login, register, getUserInfo, updateUser. As a underlying support component, its user identity and permission information are relied on by almost all upper-layer business components.

Course Management Component: Manages basic course information, including course creation, query, update, deletion, and maintenance of the association between teachers and courses. Interfaces: createCourse, getCourseList, updateCourse, deleteCourse. It is the core association component for multiple businesses such as knowledge base, lesson preparation design, test questions, and practice.

AI Intelligent Service Component: Encapsulates all interaction logic with the Alibaba Cloud Bailian (Tongyi Qwen large model), integrates RAG (Retrieval-Aided Generation) capabilities, accesses cloud-based knowledge bases, and provides a unified intelligent interface for upper-layer business components. Core classes/interfaces: AiAppService, AliyunConfig. Key technology stack: Alibaba Cloud Dashscope SDK. Key methods: Generate Subject Outline, Generate Teaching Plan, Generate Cursor, Generate Edu Lesson Plan, Generate Edu Question, Call With Session Stream, Call Alibaba Application. This component serves as the system's intelligent core engine; lesson preparation design, question management, Q&A, and subject planning components all heavily rely on its AI generation capabilities. It performs identity verification by reading configuration files and supports binding dedicated knowledge bases to enhance knowledge integration.

Teaching Resources and Lesson Preparation Components: Manages teachers' lesson preparation resources, including the creation, editing, storage, and version control of lesson plans, supporting the entire workflow from AI-generated content to manual revisions. Core Classes/ Interfaces: EduLessonPlanService,EduLessonPlanController.Key Methods:save LessonPlan, getlessonPlanByCourseId, updatelessonPlan. This component relies on the course management component to obtain course context and on the AI intelligent service component to generate initial lesson preparation content.

Question and Exam Management Component: Responsible for question bank management, exam paper generation, exam release, student answer tracking, and automatic grading, with AI-assisted question generation support. Core Classes/ Interfaces: EduQuestionService, EduExam Service, EduStudentExamService, ExamRecordController. Key Methods: GenerateQuestions, startExam, submitAnswer, autoGrade. This component relies on the Course Management Component to define exam scope, the AI Intelligent Service Component for batch question generation, and the User Management Component for student score recording.

Knowledge Base Management Component: Manages the upload and storage of local knowledge base files, synchronizes them to the Alibaba Cloud Bailian index, and supports discipline-binding with the knowledge base. Core Classes/ Interfaces: EduKnowledgeBase Service, FileController. Key Methods: uploadKnowledgeFile, bindKnowledgeToSubject, syncToAliyun. This component provides RAG retrieval data sources for AI-powered intelligent services, ensuring the accuracy and professionalism of AI-generated content.

3.5 Data Design

3.5.1 Database Selection

MySQL 8.0: Used as the primary database to store all structured business data, such as user information, course information, test questions, and practice records. MySQL features high performance, high reliability, ease of use, and open-source, well meeting the project’s ACID transaction requirements.

Alibaba Cloud Bailian Knowledge Base Service: As a vector retrieval service, it stores and processes vectorized feature data from local knowledge base documents. The project utilizes the Knowledge Base API provided by Alibaba Cloud Bailian to achieve automatic document segmentation, vectorization, and semantic retrieval without requiring a dedicated vector database. Leveraging the Embedding capabilities of the Tongyi Qwen large model, the platform enables efficient semantic-based knowledge retrieval and serves as the cornerstone for deep integration between large models and local knowledge bases.

3.5.2 Data Table Design

The data tables involved in this project follow the three normal forms of database design and are properly indexed to optimize query performance. The following are partial key table structure designs:

Table 1: User Table

Field Name	Data Type	Length	Primary Key	Not Null	Description
userID	bigint	20	Yes	Yes	User primary key ID
username	varchar	50	No	Yes	Username
password	varchar	100	No	Yes	Password (encrypted storage)
realName	varchar	50	No	No	Real name
userType	tinyint	4	No	Yes	User type (1-Administrator, 2-Teacher, 3-Student)
email	varchar	50	No	No	Email
phone	varchar	20	No	No	Phone number
createTime	datetime	-	No	Yes	Creation time
status	tinyint	4	No	Yes	Status (1-Enabled, 0-Disabled)

User table is shown in Table 1.

Table 2: Lesson Plan Table

Field Name	Data Type	Length	Primary Key	Not Null	Description
planID	bigint	20	Yes	Yes	Lesson plan primary key ID
courseID	bigint	20	No	Yes	Course ID (related to course table)
planName	varchar	100	No	Yes	Lesson plan name
knowledgePoints	text	-	No	Yes	Knowledge point content
userID	bigint	20	No	Yes	Designer user ID
createTime	datetime	-	No	Yes	Creation time
status	tinyint	4	No	Yes	Status (1-Published, 0-Draft)

Lesson plan table is shown in Table 2.

Table 3: Student Practice Table

Field Name	Data Type	Length	Primary Key	Not Null	Description
practiceID	bigint	20	Yes	Yes	Practice record primary key ID
userID	bigint	20	No	Yes	Designer user ID
questionID	bigint	20	No	Yes	Question ID (related to question table)
practiceContent	text	-	No	Yes	Student's answer content
score	decimal	10,2	No	No	Score
practiceTime	datetime	-	No	Yes	Practice time
courseID	bigint	20	No	Yes	Course ID (related to course table)

Student practice table is shown in Table 3.

Table 4: Learning Analysis Table

Field Name	Data Type	Length	Primary Key	Not Null	Description
analysisID	bigint	20	Yes	Yes	Analysis record primary key ID
courseID	bigint	20	No	Yes	Course ID (related to course table)
userID	bigint	20	No	No	Designer user ID
correctRate	decimal	10,2	No	Yes	Correct rate
analysisResult	text	-	No	Yes	Analysis result

Learning analysis table is shown in Table 4.

Table 5: Question Table

Field Name	Data Type	Length	Primary Key	Not Null	Description
questionID	bigint	20	Yes	Yes	Question ID
questionType	tinyint	4	No	Yes	1-Multiple choice, 2-Fill in the blanks, 3-Short answer, 4-Programming
questionContent	text	-	No	Yes	Question content
answer	text	-	No	Yes	Reference answer
courseID	bigint	20	No	Yes	Course ID (related to course table)
userID	bigint	20	No	Yes	Designer user ID
createTime	datetime	-	No	Yes	Creation time

Question table is shown in Table 5.

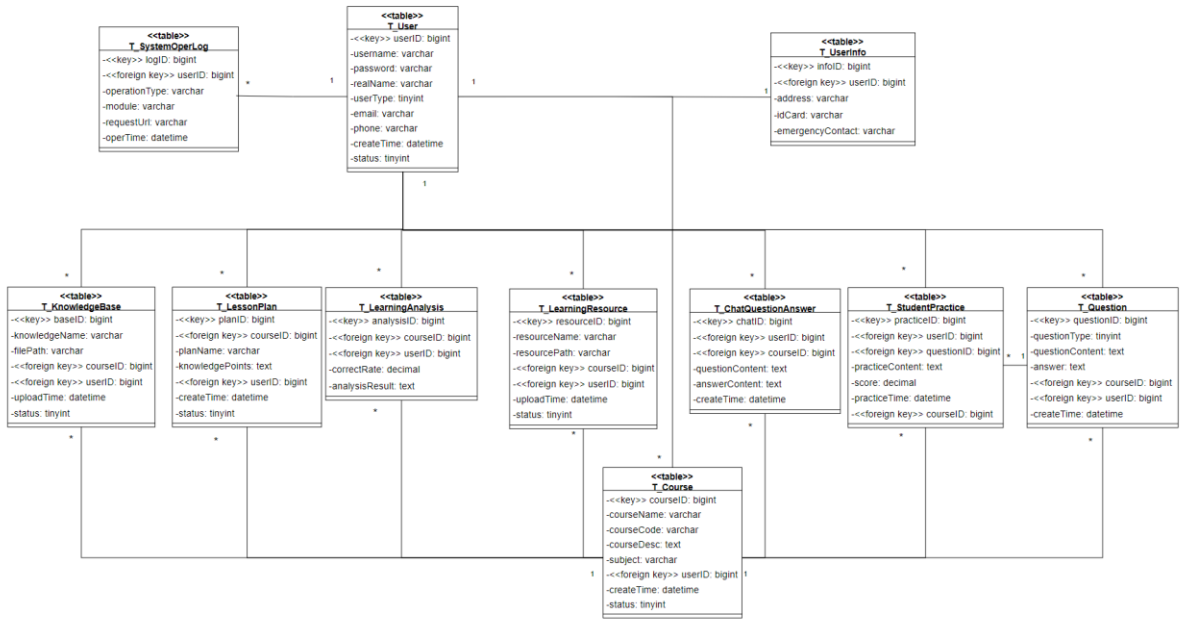

Figure 4: Database Table Structure Design Diagram

Database table structure design diagram is shown in Figure 4.

3.5.3 Data Storage

Data Storage Methods

Structured data: Managed through MySQL 8.0's InnoDB storage engine, which leverages its support for transactions, row-level locks, and foreign keys to ensure data integrity and consistency. All business data—including user information, courses, exam questions, and examination records—are stored in the MySQL database.

Unstructured Data: The original documents in the knowledge base adopt a dual storage approach, comprising local and cloud storage. Local storage saves document files in the uploads/knowledge directory of the server file system, while cloud storage synchronously uploads them to the knowledge base service on the Alibaba Cloud Bailian for management and processing. The database tables store only file metadata, including file name, path, size, MD5 value, and Alibaba Cloud file ID.

Vector data: The text-vectorized data is automatically managed and stored by the Alibaba Cloud Bailian. The project utilizes the Alibaba Cloud Bailian Knowledge Base API and the Tongyi Qwen Embedding model to perform unified automatic document segmentation, vectorization, and index construction by the platform.

3.6 Deployment Design

3.6.1 Deployment Environment Configuration

Hardware Configuration

Application Server (one unit suffices for small-to-medium deployments): CPU: 4-core or above (Intel Xeon or equivalent processor); Memory: 16–32 GB; Storage: 200 GB SSD (for system, application, and database), with optional expandable storage for knowledge base files; Network: Stable internet connection (requires access to Alibaba Cloud Bailian API).

Software Configuration

Operating System: Windows Server, Linux (CentOS 7+ or Ubuntu 18.04+) or macOS; any operating system supporting JDK 1.8. Database: MySQL 8.0 (install directly or deploy via Docker). Running Environment: JDK 1.8, Maven 3.6+, Node.js 16+, npm, and Nginx 1.18+ (optional). Cloud Service dependencies: Alibaba Cloud Bailian account (for large model APIs and knowledge base services); configure parameters such as Alibaba Cloud Access Key and WorkspaceId in application.yml.

Deployment Architecture Description:

Users access the system via the internet, with all traffic routed through the Nginx service deployed on servers. Nginx handles static resource requests and routes API requests to the backend Spring Boot application service via reverse proxy.

The application server runs the Spring Boot business application, handling core system business logic processing, local file storage, and database interactions. It ensures stable service operation through process monitoring tools.

System data is uniformly stored in the local MySQL 8.0 database, which handles all structured business data to ensure data persistence, integrity, and consistency.

AI computing power and vector processing capabilities are delivered through the Alibaba Cloud Bailian cloud platform. Local servers do not require deploying large models or vector databases; instead, they access the Alibaba Cloud Bailian API via the public network to perform core AI tasks such as document vectorization, large model inference, and knowledge base management. Data exchange between local and cloud environments is conducted using the HTTPS protocol.

3.6.2 Deployment Process and Precautions

Deployment Process:

Environmental Preparation: Install the appropriate operating system on the server and complete network configuration; install JDK 1.8 and configure the JAVA_HOME environment variable; install MySQL 8.0 database, Maven 3.6+, Node.js 16+, and npm tools; ensure the server has a stable network connection and can access the Alibaba Cloud Bailian API seamlessly.

Database Deployment and Initialization: Start the MySQL service, create a project-specific database, execute the project's built-in `edu_db.sql` script to initialize the database table structure and sample data, and specify the correct database connection details (address, username, password, etc.) in the `application.yml` configuration file.

Backend Application Deployment: Package the backend project using the Maven tool, generate an executable JAR file by skipping the testing phase, and upload the JAR package to the server. Create the upload/knowledge directory on the server and configure read/write permissions. Launch the project via the `java-jar` command with port 8290 as the default; this can be customized as needed. Additionally, configure a process daemon to ensure continuous background operation of the service.

Front-end resource deployment: Run the packaging command in the front-end project directory to generate the dist static resource directory. You can choose to place the static files into the SpringBoot static resource directory for re-packaging and deployment, or deploy them to the Nginx static resource directory. If using Nginx, configure reverse proxy rules to differentiate between static resource requests and backend API requests for routing paths.

Alibaba Cloud Service Configuration: Fully configure the Alibaba Cloud Bailian authentication parameters in application.yml, including API Key, AccessKey ID, AccessKey Secret, workspace ID, and application ID, to ensure the server can securely access Alibaba Cloud API endpoints via HTTPS.

Network and Security Configuration: Configure server firewall rules to open required ports such as 80,443, and 8290. For external HTTPS access, configure an Nginx SSL certificate. Safeguard sensitive information in the configuration files, including Alibaba Cloud keys and database passwords, and prohibit their submission to code version repositories.

Precautions:

Cloud service dependency: The system's core AI capabilities rely on the Alibaba Cloud Bailian, requiring stable server network connectivity and proper cloud API access. Regularly check the Alibaba Cloud account balance and API call quotas to prevent service interruptions. Strictly safeguard Alibaba Cloud Access Keys, Secrets, and other confidential information to avoid data breaches.

Knowledge base file management: Ensure sufficient disk space in the server's upload/knowledge/storage directories, and regularly back up knowledge base files within these directories. Strictly adhere to the Alibaba Cloud Bailian rules to control file size and format, preventing synchronization failures.

Version Consistency: The development, testing, and production environments uniformly adopt JDK 1.8 and MySQL 8.0 to ensure a consistent foundational runtime environment. The core configuration in application.yml remains identical across all environments, with environmental-specific configurations distinguished by Spring Profiles.

Log and Monitoring: Regularly review system operation logs to identify abnormal errors; continuously monitor server resource utilization including CPU, memory, and disk usage; periodically monitor API call frequency, response latency, and call exception records on the Alibaba Cloud platform.

First Launch: During the initial project launch, core components such as MyBatis Plus are automatically initialized, resulting in a brief startup delay. Before launching, ensure the database connection is valid and Alibaba Cloud configurations are correct; otherwise, the service may fail to start. After startup, verify the SDK and database connection status through logs.

Simplified deployment solution (development/test environment): The development and test environments can adopt a single-machine deployment mode, with MySQL database and Spring Boot application running on a single machine; the front-end can directly launch the development service via npm run serve.

4 Conclusions

This study has completed the design of a university practical training teaching support system that deeply integrates the Tongyi Qwen large-scale model technology from Alibaba Cloud Bailian, comprehensively covering the entire process—from intelligent lesson plan design and online Q&A to automated assessment and learning progress analysis. By leveraging cloud-based large model API services, the system enables intelligent and adaptive teaching workflows. However, the system still has several limitations: its reliance on cloud API services may introduce network latency affecting response speeds, and teacher-student data must be transmitted to the cloud for processing; within the constrained development timeframe, the accuracy of assessing complex open-ended questions remains improvable. Future research will focus on optimizing model invocation strategies to reduce latency, expanding the diversity of question types, and conducting in-depth analysis of learning data to deliver more precise and personalized learning resource recommendations.

References

[1] López L C M, Fuentes L R, Larrañaga E A, et al. School management team and inclusion: the perspective of families and teachers in the Spanish context[J]. International Journal of Leadership in Education, 2026, 29(3): 540-556.

[2] Arifin A, Mohamed R, Yusof Y, et al. Discovering Best Practices For Managing Solar Hybrid Systems In Rural Malaysian Schools[J]. IOP Conference Series: Earth and Environmental Science, 2026, 1598(1): 012020.

[3] Simwanza A. Sports activities toward sports talent development: Do the perceptions of school administrators, teachers and students matter?[J]. Social Sciences & Humanities Open, 2026, 102383.

[4] Nabus V R, Homillano M L M. Implementing School Governing Councils in Philippine Basic Education: An Analysis ofChallenges and Outcomes inDepEd Camarines Sur[J]. Journal of Scientific Research and Reports, 2025, 31(12): 535-554.

[5] Tamura H, Kawasaki H, Yamasaki S. School Management Council and Disaster Preparedness: A Study of Special Needs Schools in Japan[J]. Safety, 2025, 11(4): 108.

[6] Berkovich I. Gender stereotypes of feminine and masculine school management and leadership: examination of traits and behaviours[J]. Gender in Management: An International Journal, 2025, 40(6): 795-811.

[7] George M G, Mpiza M. Factors Influencing Parental Involvement in Schools on Improving School Management Practices: The Case Study of Selected PublicPrimary Schools in Kahama Municipality, Tanzania[J]. Asian Journal of Education and Social Studies, 2025, 51(10): 903-913.

[8] Ansari A M A, Hilal A J A, Alansari E A. Public school principals' role in supporting mathematically gifted students by equipping them with 21st-century skills[J]. Quality Assurance in Education, 2025, 33(4): 671-684.

[9] Suglo, Enoch K, Mensah A M, et al. Perceived Usefulness and Teachers' Willingness to Integrate Technology in School Management: Evidence from the Nyohini Educational Circuit, Tamale Metropolis, Ghana[J]. Asian Journal of Education and Social Studies, 2025, 51(8): 1308-1320.

[10] SAWEN N M, Maalim A, Kevogo A. Resource Allocation by School Board of Management on the Re-admission Policy for Adolescent Mothers in Tana River Sub-County PublicSecondary Schools[J]. Asian Journal of Economics, Business and Accounting, 2025, 25(7): 231-240.

[11] Tian Q. Research on the optimization path of school management from the perspective of Foucault's discipline theory[J]. Exploration of Educational Management, 2025, 3(7):

[12] Ngobeni T S. Establishing and maintaining school-community partnerships: a challenge for school management teams[J]. International Journal of Leadership in Education, 2025, 28(4): 903-928.

[13] Mekolle M P. Good Governance in School Administration and its Impact on Educational Outcome: Insights from Selected Pre-Tertiary Educational Institutions in the AnglophoneRegions of Cameroon[J]. Asian Journal of Education and Social Studies, 2025, 51(6): 1188-1208.

[14] Tupthai T, Chansirisira P. Needs Assessment of Participatory School Management for Academic Effectiveness in Provincial Administrative Organization Schools: A Study of Northeast Thailand[J]. Journal of Education and Learning, 2025, 14(5): 348.

[15] Cao L. A study of the impact of the public management concept on the reform of the modern school management system[J]. Applied Mathematics and Nonlinear Sciences, 2025, 10(1).

[16] Ngonge G R. Assessing the Participation of Local Community Members in Schools Managerial Activities and Decisions Making: A Case of Community Secondary Schools in Coast and Kagera Regions, Tanzania[J]. Asian Research Journal of Arts & Social Sciences, 2024, 22(12): 137-147.

[17] Koçak S, Sümer M. What did school administrators learn from the pandemic? A qualitative research[J]. Information and Learning Science, 2024, 125(11/12): 911-928.

[18] Atici K E, Turhan M. The impact of simulation applications on the problem-solving and decision-making skills of school administrators and prospective administrators: An experimental study[J]. Education and Information Technologies, 2024, 30(2): 1-24.

[19] Dalia B, Anit S, Ronit B. How does school management team empowerment relate to teachers' organizational citizenship behavior? A moderated mediation model[J]. International Journal of Educational Management, 2024, 38(4): 1045-1061.

[20] Sakkumpa I, Nopmanee M, Numpong J, et al. Guidelines for the Application of Design Thinking Processes in School Administration of School Administrators Under the Secondary Educational Service Area Office, Phitsanulok, Uttaradit[J]. Journal of Education and Learning, 2024, 13(1): 172.