indire International Network for Digital Innovation Research & Education

JDIH | Journal of Digital Innovation for Humanity

Innovative Question Bank Management System – Leveraging Digital Technology for Learning

Xiangjie Deng [1], Xiang He [2], Ying Zhu [3*]

Flying College, Beihang University, Beijing, China 100191
1 xiangjie_Deng@163.com
2 xiang_he@buaa.edu.cn
3 bh_zhuying@buaa.edu.cn

* Corresponding Author

– Article Type –
Original Article

{Double-Blind Peer Review}

Citation:
Deng, X.., He, X., & Zhu, Y. (2024) Innovative Question Bank Management System – Leveraging Digital Technology for Learning. *Journal of Digital Innovation for Humanity - JDIH*, Vol. 5, pp. 1-13.

https://doi.org/10.31355/149

Academic Editor(s):
Raafat George Saadé

Processing Data:
Received: June 20, 2024
Accepted: July 1, 2024
Published: September 6, 2024

Abstract: This paper introduces a web-based question bank management system (QBMS) designed for the "Ground Theory and Aviation Systems Engineering" course at Beihang University. The system, developed using Vue.js and Spring Boot, streamlines question management, practice, and online assessments. It incorporates a collaborative filtering algorithm to offer personalized practice recommendations, enhancing both teaching efficiency and student learning outcomes. The QBMS addresses the scalability issues of manual question management and introduces Spearman Correlation-based question filtering for a more adaptive learning experience. This research contributes a robust educational tool, setting a foundation for future advancements in course-specific educational technology.

Keywords: Spring Boot, Vue, Pilots, Management System, Digital, Learning, Questions Bank,

1.0 Introduction

The management of question banks in the field of flight technology is crucial for enhancing the efficiency of both teaching and learning processes. Traditionally, the management of question banks has been a manual task, with teachers or teaching assistants responsible for typing and editing questions. However, as the number of pilot cadets increases rapidly, the need for computer and network technology to manage question banks has become imperative.

A review of the current state of question bank management systems reveals that while there has been progress in the past decades, there are still issues with the quality, classification, and difficulty assessment of questions. In contrast, some research in other fields has a more solid foundation, with standardized practices established since the 1980s, providing valuable insights for the development of domestic systems (Morin et al., 2015, Ly et al., 2017, Ly et al., Saadé et al., 2011, Schmitt & Saadé, 2017).

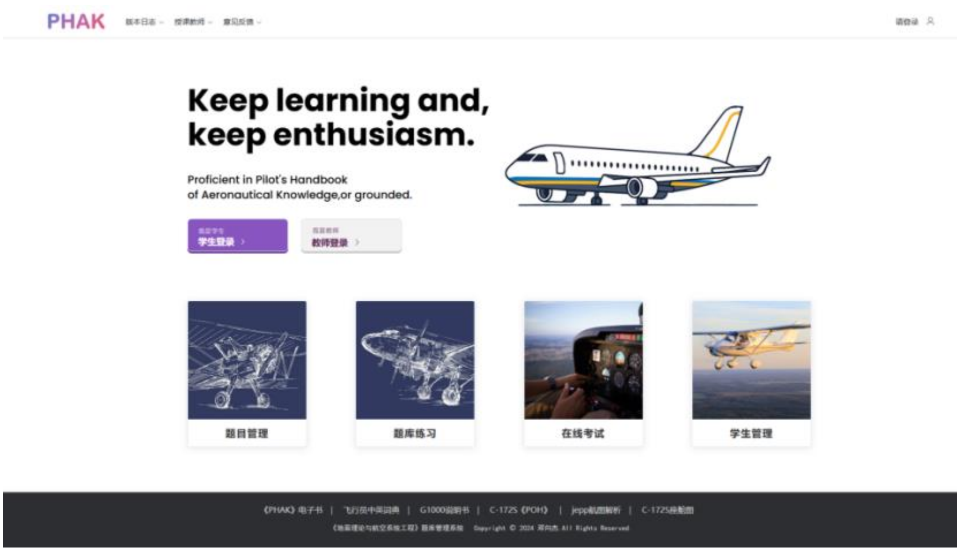

Figure 1. Homepage of the proposed online question bank management system (QBMS) for student cadets.

The primary goal of this research is to develop a question bank management system that is well-suited for the "Ground Theory and Aviation Systems Engineering" course at Flying College, Beihang University. The system aims to provide high-quality question bank resources and efficient management tools to help educators and student cadets overcome challenges in both teaching and learning (Mazin S Al-Hakeem and Mohammad Salim Abdulrahman, 2017).

The "Ground Theory and Aerospace Systems Engineering" course is one of the core courses for the Flight Technology major that covers a wide range of flight-related knowledge points. To improve the efficiency and quality of students' learning, this paper designs and implements a question bank management system aimed at assisting students in learning and practicing knowledge points in an intelligent manner. This paper analyzes the background of question bank management systems and the research progress at home and abroad, highlighting the necessity of applying modern question bank management systems to the "Ground Theory and Aviation Systems Engineering" course. The system is designed to meet the needs of the course's unique requirements, particularly the student-led question proposition teaching method. The developed system is online and available at *http://101.201.209.30/phak/*.

2.0 System Design

2.1. Main Technologies

The system utilizes several key technologies, including Spring Boot, Vue, MySQL.

Spring Boot, as a pivotal framework for backend development, has garnered significant acclaim for its streamlined approach to configuration, adhering to the "convention over configuration" philosophy. This principle simplifies the setup process by reducing the need for extensive XML configuration files, which were traditionally required in older frameworks. Instead, Spring Boot favors sensible defaults and auto-configuration, which means that developers can get their applications up and running with minimal setup.

The framework's popularity is further bolstered by its extensive ecosystem of starters, which are dependency descriptors that make it incredibly easy to include all the necessary components for a particular piece of functionality without the need to manage individual libraries. This modularity not only speeds up development but also promotes cleaner code by encouraging the use of standard practices.

Moreover, Spring Boot's integration with Actuator endpoints provides a powerful set of tools for monitoring and managing applications in production, offering insights into metrics, health checks, and more. This feature set is invaluable for maintaining the health and performance of applications in a live environment (Yingying Chen and Taizhi Lv, 2022).

Vue, a progressive framework renowned for its simplicity and flexibility, is specifically tailored for crafting dynamic and responsive user interfaces. It embraces the Model-View-View Model (MVVM) architectural pattern, which facilitates a clear separation of concerns, making the codebase easier to manage and the development process more intuitive. The MVVM pattern allows developers to focus on the view layer without worrying about the underlying data logic, as Vue handles the synchronization between the view and the model automatically.

One of the core strengths of Vue lies in its reactivity system, which efficiently tracks changes to the data and updates the DOM accordingly. This DOM monitoring capability ensures that the user interface remains in sync with the underlying data state, providing a seamless user experience. The framework's data binding features further enhance this reactivity by allowing developers to declaratively bind the DOM to the underlying data. This means that any changes to the data are instantly reflected in the UI, and vice versa, without the need for manual DOM manipulation.

Vue's component-based architecture is another key feature that makes it exceptionally well-suited for developing complex single-page applications (SPAs). Components in Vue are reusable, self-contained pieces of code that encapsulate specific functionality, making the application modular and easier to scale. This component-based approach not only promotes code reuse but also simplifies the development of large-scale applications by allowing developers to build upon a library of pre-existing components.

MySQL, a robust and widely-adopted relational database management system (RDBMS), plays a pivotal role in the backend infrastructure of numerous applications. Renowned for its efficiency, MySQL offers a high-performance database engine that is capable of handling large volumes of data with ease. Its reliability is a testament to its ability to maintain data integrity and consistency, even under the stress of concurrent transactions and high user loads.

The fast retrieval speed of MySQL is facilitated by its use of indexes, which allows for quick access to data without the need to scan entire tables. This optimization is crucial for applications that require rapid response times and are dealing with large datasets. MySQL's comprehensive search functionality further enhances its capabilities, providing developers with a rich set of tools for querying and manipulating data. This includes support for a variety of search criteria, from simple WHERE clauses to complex joins and subqueries.

MySQL's flexibility is evident in its support for various data types, ranging from integers and floating-point numbers to strings and dates, accommodating a diverse array of data storage needs. It also supports transactions, ensuring that operations are processed reliably and in an all-or-nothing manner, which is critical for maintaining data accuracy in environments where multiple operations need to be executed as a single unit.

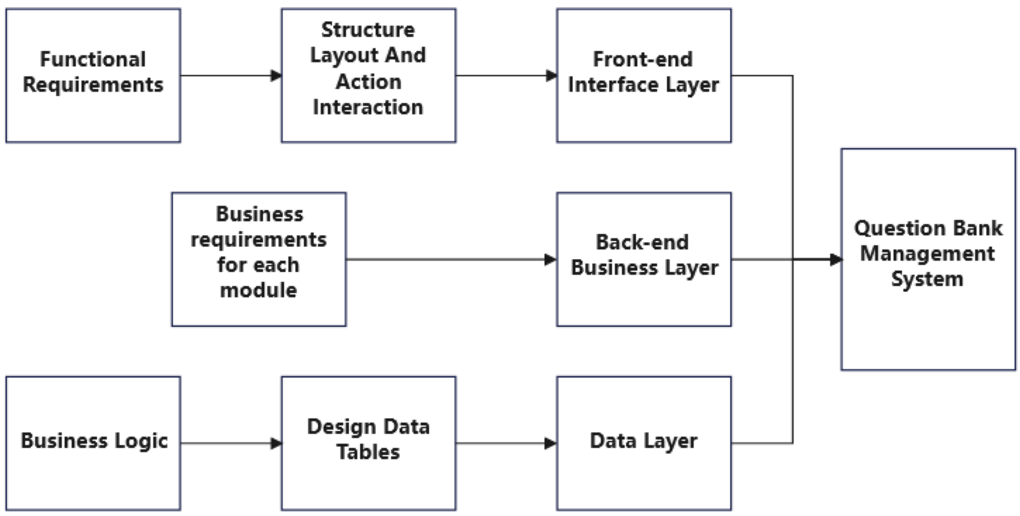

Figure 2. Overall system architecture design

By leveraging these technologies, the system provides an integrated platform supporting functions such as question upload, question bank practice, intelligent practice, and online examinations to meet the needs of students and teachers in the Flight Technology major.

2.2. System Architecture

The System Architecture section of this work meticulously delineates the structural design and functional planning of the Question Bank Management System for "Ground Theory and Aviation Systems Engineering" course at Beihang University's School of Flight. The system is developed using web technologies and is tailored to meet the specific needs of the course, emphasizing student autonomy in question creation and management.

The architecture is based on a three-tiered model, comprising the front-end interface layer, the back-end business logic layer, and the underlying database layer. The front-end utilizes the Vue framework to construct an interactive and user-friendly interface, while the back-end is powered by the Spring Boot framework to handle core business logic and data management. The MySQL database is employed for efficient data storage and retrieval.

The system is designed with a focus on modularity and scalability, ensuring that it can evolve alongside the course's development. It caters to two primary user roles: students and teachers. For students, the system offers functionalities such as login, question uploading, sequential practice, intelligent practice, and mock exams. The intelligent practice module leverages collaborative filtering algorithms to provide personalized question recommendations based on students' historical performance and question similarities.

For teachers, the system includes modules for question editing, final exam compilation, and student management. Teachers can efficiently manage and review questions, generate sample tests quickly according to the teaching schedule, and effectively manage student information. The test exam compilation module is designed to produce both teacher and student versions of the exam papers to meet diverse educational and assessment needs.

The database design begins with a conceptual model using an Entity-Relationship Diagram (E-R Diagram), which is then translated into a logical model involving the creation of database tables, fields, and relationships. This design approach minimizes data redundancy and enhances data management efficiency.

In summary, the system architecture is robust, efficient, and user-centric, providing a comprehensive solution for question bank management that enhances teaching and learning efficiency while ensuring data security and system maintainability. The system's design not only optimizes the utilization of educational resources but also supports the teaching management with a solid foundation, considering data maintainability and scalability (SR Bharamagoudar, RB Geeta, and SG Totad, 2013).

2.3. Functional Design

The system supports user authentication and permission management. Once registered and logged in, students can use the system to upload questions. The student homepage provides a functionality called "Upload Questions," allowing students to contribute their own questions to the Ground Theory Question Bank. The homepage also offers a "Question Bank Practice" feature, allowing students to view available question banks. The page displays information about each question bank, including its name, the number of questions, and the number of completed questions. Additionally, students can engage in "Smart Practice," which intelligently generates practice questions tailored to their commonly incorrect answers and weak knowledge areas. The "Question Search" feature enables students to search for complete questions, answers, and explanations by entering partial content or keywords. Students can also participate in online mock exams through the "Online Examination" section.

Upon logging into the system, teachers can access the "Edit Questions" section to modify existing questions, including changing question details or deleting questions. In the "Question Review" section, teachers can review the questions uploaded by students.

In the "Exam Management" section, teachers can create question sets for exams based on chapter proportions. They can export these question sets as Word documents or PDFs. The "Student Management" section allows teachers to add, delete, modify, and query student information (Md Monir Ahamed, 2019).

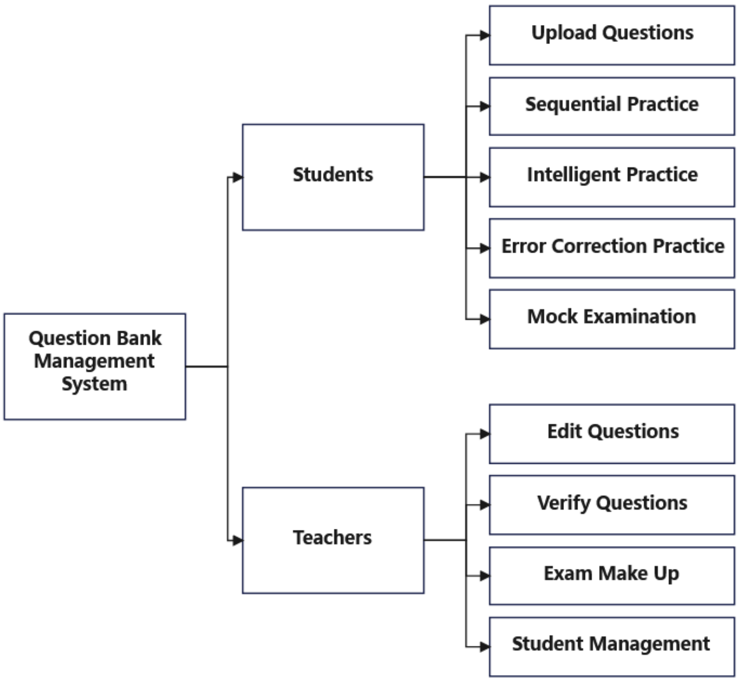

Figure 3. Overall functional module design of the system

3.0 System Implementation

3.1. Back End Implementation

The backend of the question bank management system is meticulously architected using the Spring Boot framework, a powerful tool within the Java ecosystem that simplifies the bootstrapping and development of new Spring applications. The framework's standout feature is its automatic configuration, which greatly reduces the complexity involved in setting up a backend server. By including an embedded Tomcat server, Spring Boot allows for the quick deployment of the system, whether in a local or cloud environment, without the need for extensive configuration (Yichun Tang, 2024).

Dependency management is streamlined through Spring Boot's starter dependencies, which are pre-configured sets of libraries that an application needs to function. This approach minimizes the effort required to add new functionalities to the system, such as database connectivity and JSON processing, by automatically including the necessary dependencies.

The backend is meticulously segmented into distinct layers, each with its specific responsibilities. The business layer is where the core logic of the application resides, handling operations such as user authentication, data validation, and business rule enforcement. The data access layer interacts with the MySQL database, performing CRUD operations to store and retrieve information, such as question banks, user data, and examination records.

One of the key enhancements provided by Spring Boot is the support for RESTful API services, which are used to facilitate seamless communication between the front-end Vue.js application and the backend services. These APIs are designed to be lightweight and are built using Spring MVC annotations, which further simplify the development process.

To bolster security, the backend implements custom annotations like @TokenRequired, ensuring that all requests are authenticated by validating tokens and other critical information. This token-based authentication mechanism safeguards the data and prevents unauthorized access.

Configuration management is centralized and simplified through the use of application.yml, a single configuration file that houses all the necessary settings for the backend, including the server's port number, database connection details, and caching strategies. This unified approach to configuration not only eases the maintenance process but also enhances the system's adaptability to different environments.

In essence, the backend developed with Spring Boot serves as a robust foundation for the question bank management system. It encapsulates the business logic, data management, and service layer functionalities, providing a scalable, secure, and efficient platform that supports the educational goals of the system. The framework's emphasis on convention over configuration, along with its robust ecosystem, ensures that the backend is not only easy to develop and maintain but also capable of evolving with the growing needs of the educational landscape.

3.2. Front End Implementation

The front end of the question bank management system is architected using the Vue framework, a progressive and flexible approach that delivers an interactive and dynamic user experience. Vue's reactive components form the backbone of the system's user interface, allowing for a seamless and intuitive interaction between the user and the application. By leveraging Vue's data binding capabilities, the system ensures that the UI remains synchronized with the underlying data model, facilitating real-time updates and reflections of user actions.

The front end is designed with a component-based architecture, which promotes reusability and modularity. Each component encapsulates specific functionality, such as question display, option selection, and feedback provision, contributing to a clean and organized codebase. Vue's single-file components encapsulate the template, logic, and style, making the development process more manageable and the components more portable (Junhui Song, Min Zhang, and Hua Xie, 2019).

The system utilizes Vue Router for routing, which enables the creation of a single-page application (SPA) that loads a single HTML page and dynamically updates the content as the user navigates through the application. This approach avoids the need for page reloads, providing a fluid and responsive navigation experience.

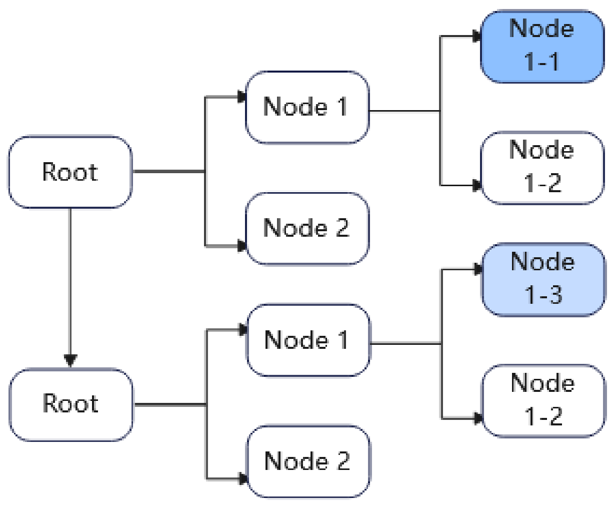

Figure 4. Front end architecture

State management is efficiently handled by Vuex, which serves as a centralized store for all the components in the application. It manages shared states such as user authentication status, question bank data, and user progress in an organized and predictable manner. Vuex's strict mutation system ensures that state changes are tracked and debugged easily, maintaining the integrity of the application's state throughout user interactions (Mengdie Gu et al, 2022).

The front end also integrates with the back end through API calls, using Vue's AJAX utilities to fetch, manipulate, and display data. The use of HTTP requests allows the front end to communicate with the back-end services effectively, fetching necessary data for operations like question uploading, retrieving questions for practice, and submitting exam results.

In addition, the front end implements a responsive design, ensuring accessibility and functionality across various devices and screen sizes. Vue's adaptability to different form factors is achieved through a combination of responsive design principles and Vue's own flexibility in rendering components based on the available screen real estate.

The intelligent practice module stands out as a unique feature, utilizing a recommendation algorithm that analyzes students' historical performance data to provide personalized question recommendations. This module enhances the learning experience by focusing on areas where students need improvement, leveraging the power of data-driven insights to tailor the practice sessions.

Security is also a key consideration in the front end's design. Vue's built-in security features, such as protection against cross-site scripting (XSS) attacks, are employed to safeguard the application and its users. Additionally, the front end ensures that all sensitive data exchanged with the back end is transmitted securely using HTTPS, protecting the integrity and confidentiality of the data.

In conclusion, the front end of the question bank management system, powered by Vue, offers a robust and engaging platform for both students and teachers. It provides a seamless, interactive, and secure environment that caters to the educational needs of its users, supported by a modern and maintainable codebase that is designed to evolve with the changing demands of educational technology.

3.3. DataBase Implementation

The database layer of the question bank management system is underpinned by MySQL, a widely-adopted relational database management system (RDBMS) known for its reliability and efficiency. MySQL serves as the foundational repository for storing, retrieving, and managing the vast array of data that propels the system's functionalities (Bhupest Rawat, Suryari Purnama, et al, 2021).

At the core of the database design is the normalization of data to eliminate redundancy and ensure integrity. The system's schema is meticulously crafted using MySQL's capabilities to define tables, columns, and relationships that accurately reflect the entities and processes of the question bank management system. This includes tables for storing questions, user data, chapter information, completion records, and examination details.

MySQL's powerful querying capabilities are harnessed to support complex searches and data retrieval operations, enabling the system to efficiently access and manipulate educational content. The system leverages SQL queries to perform operations such as selecting questions based on categories, retrieving user progress, and generating reports on examination results.

The database is also designed to support transactions, which ensures that operations such as updating user scores, recording completion times, and logging activity are performed reliably and atomically. This is critical for maintaining the accuracy and consistency of the data, especially in operations that involve simultaneous updates from multiple users.

To enhance performance, MySQL's indexing features are utilized to optimize the retrieval of data, ensuring that searches and queries are executed with speed and precision. The system takes advantage of indexes on frequently searched fields such as question IDs, user IDs, and chapter names to expedite the lookup process.

Security is a paramount concern, and MySQL provides robust mechanisms to safeguard the data. The system implements strict access controls, encryption of sensitive data, and regular audits to protect the integrity and privacy of the information stored within the database.

Furthermore, MySQL's scalability and flexibility are instrumental in accommodating the growing needs of the system. As the number of users and the volume of data expand, MySQL's ability to handle increased loads and its support for large datasets ensure that the system remains responsive and efficient.

The system also employs MySQL's backup and recovery features to ensure data durability and availability. Regular backups are scheduled to protect against data loss, while recovery procedures are in place to ensure business continuity in the event of a system failure (Ying Zhang and Feng Pan, 2022).

In summary, MySQL plays a pivotal role in the question bank management system by providing a robust, scalable, and secure platform for data storage and management. Its comprehensive feature set supports the complex requirements of the system, from data integrity and security to performance optimization and scalability, thereby laying a solid foundation for the system's operation and growth (Sunghoi Park et al, 2013).

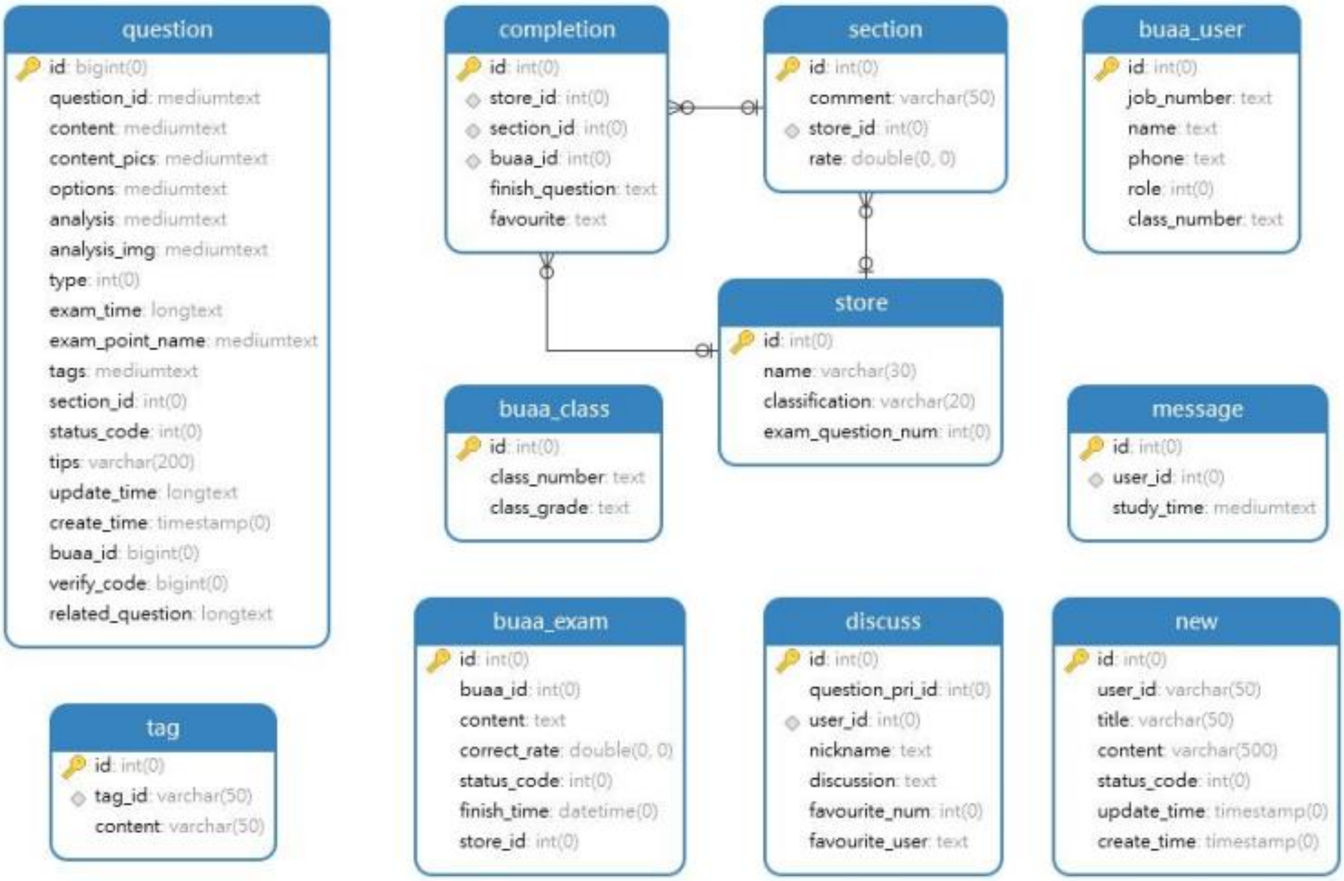

Figure 5. Database Table E-R Diagram

4.0 System Implementation and Testing

This section provides an overview of the implementation and testing of the question bank management system, highlighting the key modules and their functionalities.

4.1. System Architecture

The system is designed with a front-end and back-end separation approach, utilizing the Vue framework for the front-end and Spring Boot for the back-end. This architecture allows for efficient management of question banks and provides a user-friendly interface for both students and teachers.

4.2. Student Role Module Implementation

The student role module encompasses several key features, including login, question uploading, sequential practice, intelligent practice, and mock exams.

The login module ensures secure access to the system by validating user credentials. Upon successful login, students are granted access to various practice and examination features.

Students can contribute to the question bank by uploading new questions, which are subject to teacher review and approval. This feature fosters an interactive learning environment and enhances the diversity of the question bank.

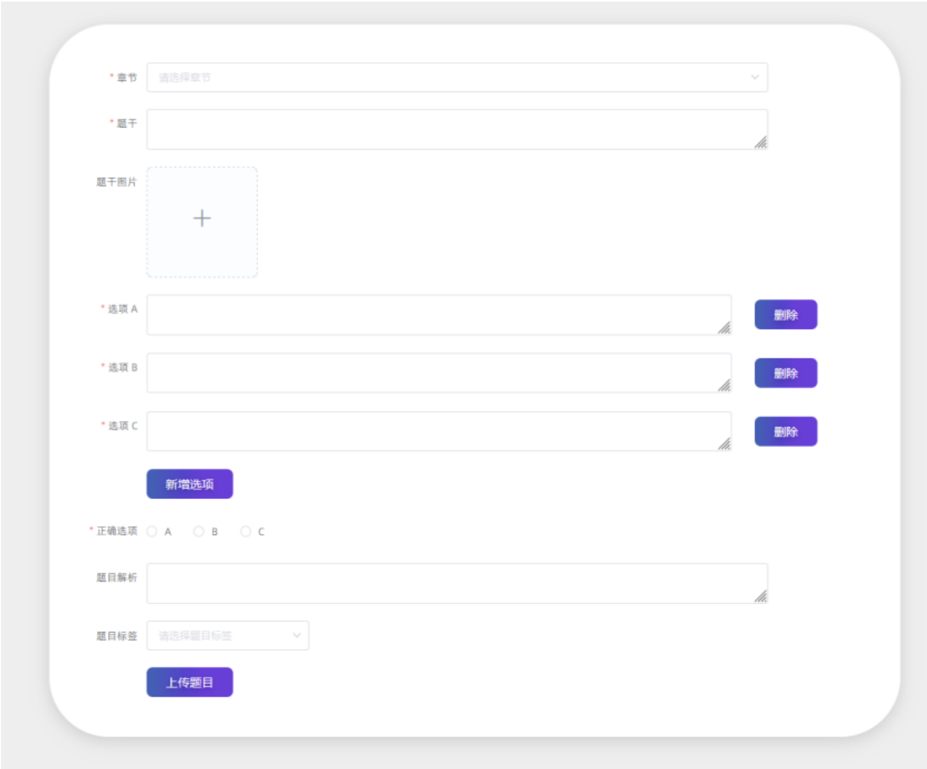

Figure 6. Question upload page

4.2.1 Sequential Practice

The practice modules allow students to engage with questions in a structured manner. The intelligent practice module leverages collaborative filtering algorithms to offer personalized question recommendations based on students' performance and question similarities.

Sequential practice is a fundamental feature of the system that allows students to work through questions in a predefined order, typically following the structure of the course material. This approach ensures that students progress through the content in a logical sequence, facilitating a comprehensive understanding of the subject matter.

Upon logging into the system, students are presented with a chapter overview. They can select a chapter to begin sequential practice, where they are guided through the questions in a systematic manner. The interface provides a clear indication of progress, allowing students to track their completion of each section.

The system maintains a record of the student's progress, including the number of questions attempted and the correctness of their answers. This information is saved locally and can be retrieved at any time, enabling students to resume practice from their last session.

4.2.2 Intelligent Practice

The intelligent practice module represents an advanced feature of the system, offering personalized question recommendations based on the student's performance and the similarity between questions.

At the core of the intelligent practice module is a collaborative filtering algorithm that analyzes student responses and identifies areas requiring further attention. The system calculates the similarity between questions using the Spearman rank correlation coefficient, which measures the monotonic relationship between the ranks of two variables.

[1] $$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where ρ is the Spearman rank correlation coefficient, d_i is the difference in ranks for each pair of questions, and n is the number of questions.

Based on the calculated similarity, the system generates a list of recommended questions that are similar to those the student has previously answered incorrectly. This targeted approach allows students to focus on areas of weakness and enhances the effectiveness of their study sessions.

In the intelligent practice module, students are presented with a set of questions curated specifically for them. The interface is designed to be intuitive, with options to review questions, receive immediate feedback, and understand the concepts more deeply through focused practice.

4.2.3 Mock Exams

The mock exam module simulates real exam conditions, providing students with a platform to assess their readiness for the actual exams. The system's automated scoring mechanism ensures immediate feedback on performance. The teacher role module facilitates the management of question banks, examination papers, and student information. Teachers can edit and review questions uploaded by students, ensuring the quality and accuracy of the content within the question bank. The system enables teachers to quickly generate exam papers based on specific criteria, such as question count and chapter distribution. This feature streamlines the process of creating assessments and contributes to efficient teaching practices.

4.2.4 Student Management

The student management module provides teachers with tools to manage student information effectively. Teachers can perform operations like adding, updating, and deleting student records as needed.

4.3. System Testing

Comprehensive testing was conducted to ensure the system's functionality aligns with the design specifications. Testing covered all modules, including user interface, business logic, and database operations. The tests confirmed the system's stability and reliability, with all modules performing as expected.

5.0 Discussion on the Design

The system architecture depicted in figure 2 represents a structured approach to designing a Question Bank Management System. This architecture integrates multiple layers and components, each addressing different aspects of the system's functionality and operational requirements ready for today's digital world centered on artificial intelligence. We explore some of the most important advantages, innovative aspects, and unique features of this architecture:

5.1 Advantages of the System Architecture

Modular Design:

The architecture separates different functionalities into distinct layers, such as the Front-end Interface Layer, Back-end Business Layer, Data Layer, and the Question Bank Management System itself. This modular design facilitates easier maintenance, updates, and scalability, allowing each module to be developed, tested, and deployed independently.

Clear Separation of Concerns:

By dividing the architecture into specific layers—like separating the Front-end Interface Layer from the Back-end Business Layer—the system ensures a clear separation of concerns. This improves the manageability of the system and reduces the likelihood of errors that could arise from overlapping functionalities.

Scalability and Flexibility:

The architecture is designed to be scalable, allowing for the easy addition of new features or expansion of existing ones without overhauling the entire system. This is particularly beneficial in the aviation education sector, where regulatory changes or new training requirements might necessitate frequent updates to the question bank.

Enhanced Data Management:

The presence of a dedicated Data Layer, where data tables are designed and managed, allows for efficient handling of large volumes of data. This is crucial in aviation education, where the question bank must cover a wide range of topics and maintain accuracy and relevance.

Improved User Experience:

The Front-end Interface Layer is designed to handle the structure, layout, and action interactions, ensuring that the user interface is intuitive and user-friendly. This is essential for enhancing the user experience, whether the users are students, instructors, or administrators.

Dynamic Business Logic Handling:

The Back-end Business Layer deals with business requirements for each module, incorporating dynamic business logic that can adapt to different educational scenarios. This flexibility is key in aviation training, where different modules might have varying requirements for questions, such as multiple-choice, simulation-based, or scenario-based questions.

5.2 Innovative Aspects of the Architecture

Integration with Business Logic and Requirements:

The architecture innovatively integrates business logic directly into the system, allowing for real-time adjustments based on evolving business requirements. This adaptability is particularly useful in aviation education, where curriculum standards and training protocols can change frequently.

Feedback-Driven Design:

By incorporating a direct feedback loop from the Front-end Interface Layer back to the Back-end Business Layer and the Question Bank Management System, the architecture allows for continuous improvement based on user interaction and feedback. This is innovative in ensuring that the system remains responsive to the needs of both educators and learners.

Customizable and Extendable Structure:

The architecture's design allows for customization and extension, meaning new modules or features can be integrated without disrupting existing functionality. This is especially valuable in aviation education, where new learning technologies or regulatory requirements may emerge.

5.3 Unique Features of the Architecture

Dedicated Question Bank Management System:

Unlike generic educational platforms, this architecture includes a dedicated Question Bank Management System that specifically addresses the needs of aviation training. This system is designed to manage complex question structures, including scenario-based questions that are crucial for pilot training and air traffic control education.

Support for Complex Data Interactions:

The architecture supports complex data interactions between the Data Layer and other components, ensuring that the system can handle the diverse and detailed data requirements typical of aviation education. This includes managing detailed metadata for questions, tracking student performance data, and ensuring data integrity across the system.

Focused on Regulatory Compliance:

The system is uniquely positioned to support the stringent regulatory compliance requirements of aviation training. The architecture's flexibility allows it to adapt quickly to new regulatory changes, ensuring that the question bank remains up-to-date with the latest industry standards and practices.

The proposed system architecture for a Question Bank Management System is particularly advantageous for the aviation education sector due to its modular design, scalability, and clear separation of concerns reflected by the diverse type of issues in the aviation sector from technical of climate change and regulations (Guan et al., 2022, 2023). Its innovative aspects lie in its ability to integrate dynamic business logic, provide a feedback-driven design, and offer a customizable and extendable structure. The architecture's unique features, such as a dedicated question bank management system and support for complex data interactions, make it well-suited to meet the specific demands of aviation education, providing a robust, flexible, and future-proof solution.

6.0 Conclusion

In the current era of rapidly advancing information technology and educational informatization, the design and implementation of question bank management systems have become a hot topic in the field of education. Despite numerous achievements, there are still gaps in existing systems when it comes to building a question bank for specific courses such as "Ground Theory and Aviation Systems Engineering." Most of these systems do not allow students to upload questions themselves, which fails to meet the personalized needs of this course. Moreover, commercial exercise software also falls short in utilizing students' incorrect answer data for question recommendations. Most importantly, the need for a smart innovative question bank management system in the area of aviation is great as the issues today extend beyond systems engineering as reported and described by Guan et al., (2022 and 2023).

In response to these issues, this study has designed and implemented a question bank management system based on Web technology, employing the Vue and Spring Boot frameworks. This software not only meets the needs of teachers in "Ground Theory and Aviation Systems Engineering" for question bank construction and management, as well as the teaching design of student-independent question setting, but also provides students with a convenient platform for question practice. Before the design, in-depth communication with course teachers was conducted to clarify the design of the system, which was then translated into code implementation. Moreover, the question bank management system adopts a front-end and back-end separation development model, which greatly reduces the system's coupling and provides great convenience for future functional expansion and maintenance.

For end users, the system offers the function of practicing questions by chapters and intelligently analyzes incorrect answer data and question similarity to recommend relevant questions to students, thereby strengthening the mastery of knowledge points. For teacher users, the system has designed a convenient question review mechanism, which greatly reduces the workload in the early stage of question bank construction. At the same time, teachers can freely choose chapters for test paper generation and assembly according to the teaching progress. In addition, the system also includes auxiliary functions such as student management.

The author hopes that this system will play an important role in the construction of the question bank for the "Ground Theory and Aviation Systems Engineering" course. As educational informatization continues to deepen, this system will continuously adapt to new educational needs and provide ongoing support for the teaching of flight technology majors.

Acknowledgment

This work is sponsored by the educational reform project from the Flying College, Beihang University.

References

1. Bhupest Rawat, Suryari Purnama, et al. “MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung” . In: International Journal of Cyber and IT Service Management 1.2 (2021), pp. 173–179.
2. Guan, H., Liu, H., & Saadé, R. G. (2023). International Aviation Smart City Alliance as a global strategy for carbon emissions reduction. International Journal of Sustainable Aviation, 9(2), 162-189.
3. Guan, H., Liu, H., & Saadé, R. G. (2022). Analysis of carbon emission reduction in international civil aviation through the lens of shared triple bottom line value creation. Sustainability, 14(14), 8513.
4. Junhui Song, Min Zhang, and Hua Xie. “Design and implementation of a vue. js-based college teaching system” . In: International Journal of Emerging Technologies in Learning (Online) 14.13 (2019), p. 59.
5. Ly, S. L. S., Saade, R. G., & Morin, D. (2017). Immersive learning: Using a web-based learning tool in a phd course to enhance the learning experience. Journal of Information Technology Education. Research, 16, 227.
6. Mazin S Al-Hakeem and Mohammad Salim Abdulrahman. (2017). Developing a new e-exam platform to enhance the university academic examinations: The case of Lebanese French University. International Journal of Modern Education and Computer Science vol. 9, No. 5, p. 9. Modern Education and Computer Science Press.
7. Md Monir Ahamed. (2019) Development of a web-based question bank for automated question paper generation. Institute of Information and Communication Technology
8. Mengdie Gu et al. (2022) Research on copyright appointment registration microplatform system based on vue. MATEC Web of Conferences. vol. 355, No. 11, p. 02036. EDP Sciences.
9. Morin, D., Thomas, J. D., & Saadé, R. G. (2015). Fostering problem-solving in a virtual environment. Journal of Information Technology Education. Research, 14, 339.Shi, Y., Ma, G., Ren, J., Zhang, H., & Yang, J. (2022). An end-to-end conformer-based speech recognition model for mandarin radiotelephony communications in civil aviation. In Proceedings of the Chinese Conference on Biometric Recognition (pp. 335-347). Springer.
10. Saadé, R. G., Buyukkurt, M. D., & Alkhori, C. (2011). Technology mediated learning: Observations in two technologies. Issues in Informing Science and Information Technology, 8, 395-408.
11. Schmitt, U., & Saadé, R. G. (2017, June). Taking on opportunity divides via smart educational and personal knowledge management technologies. In *Proceedings of the 12th International Conference on e-Learning (ICEL)* (pp. 188-196).
12. SR Bharamagoudar, RB Geeta, and SG Totad. (2013). Web based student information management system. International Journal of Advanced Research in Computer and Communication Engineering. vol. 2, No. 6, pp. 2342– 2348.

13. Sunghoi Park et al. “Design and implementation of smart energy management system for reducing power consumption using ZigBee wireless communication module” . In: Procedia Computer Science 19 (2013), pp. 662–668.
14. Yingying Chen and Taizhi Lv. (2022). Design and Implementation of COVID-19 Data Analysis and Visualization System. Scientific Journal of Technology vol. 4, No. 11.
15. Yichun Tang. “Design and Implementation of a Student Management System Based on Springboot Framework Technology” . In: ICSETPSD 2023: Proceedings of the First International Conference on Science, Engineering and Technology Practices for Sustainable Development, ICSETPSD 2023, 17th-18th November 2023, Coimbatore, Tamilnadu, India. European Alliance for Innovation. 2024, p. 301.
16. Ying Zhang and Feng Pan. (2022). “Design and implementation of a new intelligent warehouse management system based on MySQL database technology”. In: Informatica 46.3.