

Design and Implementation of Monitoring Education Collaboration Framework

Yafei Zhu[#], Yiping Zhang[#], Bingjie Li, Zihao Chen^{*}

Department of Mathematics, Hebei GEO University, Shijiazhuang City, Hebei Province 052161, China

^{*}Corresponding author: Zihao Chen (Email: 13474605980@163.com)

[#]These authors contributed equally.

Abstract: This paper delves into the design and implementation details of a monitoring-education collaborative framework, addressing the integration challenges of real-time device monitoring and personalized skill development. Focusing on the robust backend architecture, the system employs MyBatis-Plus for efficient data persistence, Spring Boot for application orchestration, and Oracle 11g for data management. Key implementation aspects include a meticulously designed data access layer with optimized transaction management and connection pool configuration, comprehensive database performance monitoring and SQL optimization strategies, a Vue.js-based frontend architecture for role-specific interfaces, and a multi-layered security control mechanism integrating RBAC and ABAC. The implemented framework provides a stable, efficient, and secure technical foundation, enabling the practical realization of the monitoring-education collaboration concept and offering concrete solutions for building enterprise-level operation and maintenance support systems.

Keywords: Monitoring education collaboration, Java system, Data access layer, Performance optimization, Safety control.

1. Introduction

The conceptual integration of monitoring and education presents significant technical implementation challenges, particularly in ensuring system stability, data handling efficiency, and operational security during the development of a collaborative framework [1, 2]. While the overarching goal is to enhance operational efficiency and personnel skills, the practical value of such a system is heavily dependent on its technical design and implementation quality.

This paper shifts the focus from the conceptual model to the detailed design, implementation, and optimization of the monitoring-education collaboration framework. It elaborates on the specific technical selections and architectural decisions that materialize the concept. We detail the construction of the data access layer using MyBatis-Plus, the implementation of transaction management and database connection pooling, and the strategies for SQL performance optimization and real-time system monitoring. Furthermore, the design of the front-end componentized architecture and the implementation of a refined, multi-layered security control mechanism are discussed. By presenting these concrete design and implementation aspects, this work aims to provide a reliable technical blueprint and practical reference for developing similar complex enterprise application systems that require high performance, stability, and security.

2. Related technologies and theories

In this project, in order to achieve efficient data center operation and maintenance monitoring system and skills improvement service function[2], this paper selects a stable and comprehensive development environment and tools. The project is developed and run based on the operating system

above Windows 8, which can meet the deployment requirements of enterprise software.

Microsoft Visual Studio 2013 is selected as the integrated development tool in the development environment. Its powerful code editing, debugging and project management functions can support the rapid development of complex systems[3]. Oracle 11g is used for database management. With its high reliability, distributed processing ability and perfect security mechanism, it provides stable data storage and efficient data management support for the system[4].

The running support environment relies on Chrome browser v69.0. With its fast page rendering ability and wide compatibility, it ensures the fluency and stability of users accessing the system through the browser. The programming language is Java, which has the characteristics of cross platform, object-oriented and high security, and is widely used in enterprise application development[5]. The system has 14020 lines of source programs, which provides a solid technical foundation for the realization of the functions of the data center operation and maintenance monitoring system, including online programming learning courses, information management of the operation and maintenance framework, etc[6].

In terms of hardware environment, core i5 quad core processor and Galaxy GTX graphics card are used for development and operation, which can meet the performance requirements of data processing and graphical display. The system is oriented to the field of data center operation and maintenance monitoring management[7]. Through powerful information management module and sound network security mechanism, it effectively supports the comprehensive service objectives of data maintenance, display and user skills improvement. The development environment and tool configuration of the paper are shown in Table 1.

Table 1: Development environment and tool configuration table

Classification	Content
Environment	Core i5 quad core processor with Galaxy GTX graphics card
System	Windows 8 up
Tool	vs2013, ORACLE11g
Support	Chrome browser v69.0
Programme	Java

3. Data access layer design

3.1. Technology selection and framework integration

Mybatis plus is enhanced on the basis of mybatis, which significantly improves the development efficiency by automating crud operation and code generation mechanism. In the process of framework integration, we first introduced mybatis plus and MySQL driver dependency through Maven to ensure that the project can establish a connection with the database and execute SQL operations[8]. Secondly, complete the data source and mybatis plus parameter configuration in the spring boot configuration file, including database connection URL, user name, password, mybatis mapping file path and type alias settings. Through these configurations, the system can realize the automatic mapping of database fields and Java entity class attributes, avoid writing a large number of conversion codes manually, and turn on the SQL log output function, which is convenient for debugging and problem positioning in the development process[9].

3.2. Interface design specification

Mapper interface design follows the naming specification of "domain model+function module", and each interface corresponds to a specific business domain data operation. For example, usermapper is responsible for adding, deleting, modifying and querying data related to user accounts, and studyrecordmapper focuses on learning record management. The interface method design combines the actual requirements of the system, including both the basic crud operation (automatically provided by mybatis plus) and the definition of complex business query methods. For example, the gettotalstudytimebyperiod method in studyrecordmapper can count the learning duration according to the user ID and time period[10]. The selectpagewithsubject method supports paging query of learning records and association of subject information to meet the system's needs for multi-dimensional retrieval of learning data. XML Mapping file is the core carrier for mybatis plus to bind SQL statements with Java methods. On the one hand, the mapping relationship between database fields and entity class attributes is configured through the resultMap tag to solve the difference between database naming conventions (such as underline naming) and Java hump naming, so as to ensure correct data conversion. On the other hand, for complex queries, such as the joint query of learning records and subject information involving multi table Association, the object relationship mapping is

realized through the association and collection tags, and the query results are encapsulated into Java object structures that meet the business requirements, providing structured data for the upper business logic.

3.3. Transaction management and connection pool configuration

The system uses spring declarative transaction management, and achieves atomic control of database operations by adding @transactional annotation to business methods. For example, when inserting learning records in batches, the transaction annotation ensures that multiple insert statements are either successfully submitted or rolled back when exceptions occur to avoid data inconsistency. Specify rollback for=exception.class in the transaction configuration to ensure that all runtime exceptions trigger transaction rollback and ensure data integrity. Hikaricp is selected as the database connection pool, which is famous for its high performance and low resource consumption. During the configuration process, the minimum idle and maximum pool size parameters are set to accurately control the minimum and maximum connections of the connection pool to avoid waste or shortage of connection resources. At the same time, idle timeout and Max lifetime parameters are configured to realize the automatic recycling of idle connections and the management of connection life cycle, so as to prevent the connection failure caused by long-term use. The connection test query configuration of the connection pool is used to regularly check the connection availability to ensure that the system always uses valid connections to access the database.

4. Performance optimization and monitoring

4.1. Database monitor

In terms of SQL performance optimization, first of all, according to the business query requirements, a composite index is created on the user_id, start\time, subject\ID and other common query fields of the study\record table, which greatly improves the efficiency of multi condition query. Secondly, through the performance monitoring plug-in of mybatis plus, set the slow SQL threshold (such as 500ms), and automatically record the SQL statements whose execution time exceeds the threshold. These records provide a basis for subsequent SQL optimization. Developers can adjust query logic or index structure to further improve database access performance.

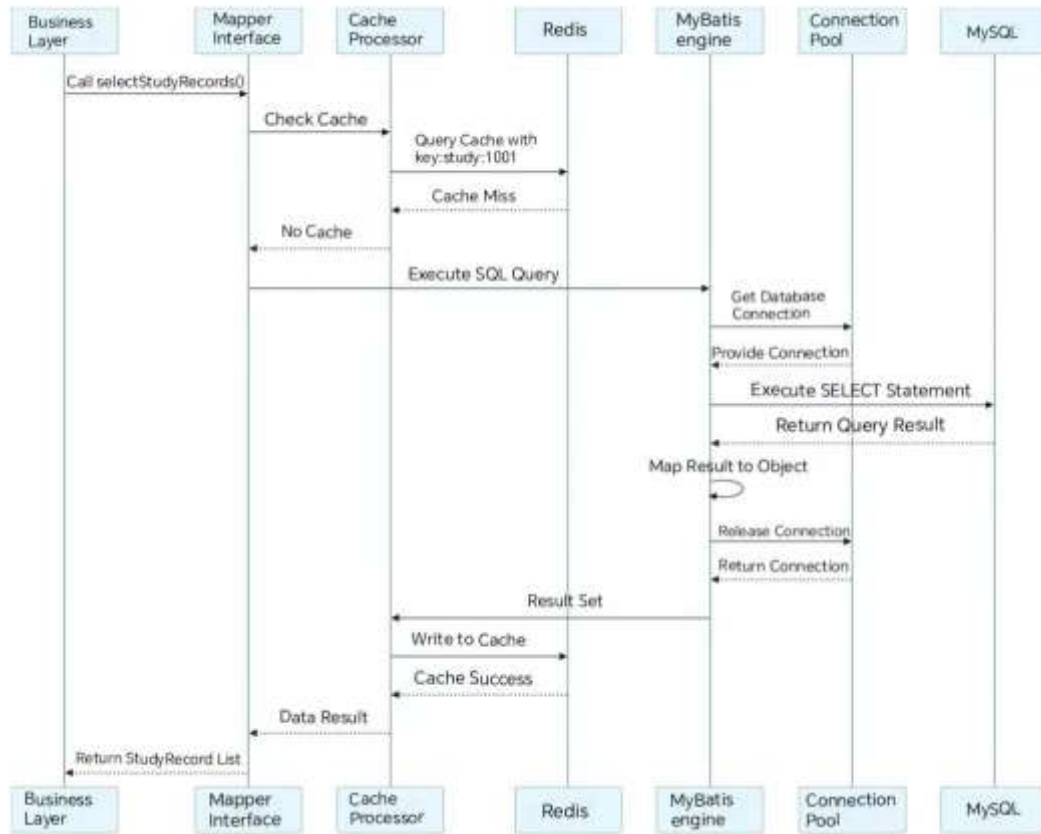


Figure 1. Database timing interaction diagram

Figure 1 shows the Database timing interaction diagram. With the monitoring function provided by spring boot actuator, the system realizes the real-time monitoring of the database connection pool. Through the/actor/datasource endpoint, the operation and maintenance personnel can obtain the key indicators of the connection pool, including the number of active connections, the number of waiting connections, the maximum number of connections, etc. These monitoring data help to timely find potential problems such as connection leakage and connection pool depletion, and ensure the stable operation of the system in high concurrency scenarios.

4.2. Front end page architecture and layout

The presentation layer uses vue.js framework for component-based development, follows the principle of modular design, and disassembles the system interface into multiple independent and reusable components to improve development efficiency and code maintainability. The overall page layout is differentiated according to user roles and functional requirements:

Student side: the main interface is presented in the form of a dashboard, using a simple and intuitive card layout. The core information, such as the learning duration of the day, the learning trend of the week, and the tasks to be completed, is displayed on an independent card to facilitate students' rapid access to key learning data. At the same time, set up a quick entry link to the learning statistics Kanban, summary file preview and other pages.

Management side: the interface layout emphasizes the comprehensiveness of functions and the convenience of operation, and adopts the classic layout mode of sidebar navigation combined with the main content area. The sidebar lists the user management, system settings, data statistics and

other core function menus. The main content area displays the corresponding details and operation interface according to the selected function.

4.3. Safety control

When a user initiates an access request, the system first performs security detection. This stage mainly analyzes the legitimacy of the request source, including verifying the user agent header information, checking the integrity of necessary parameters (such as sessionid or token), and analyzing the traffic characteristics to prevent malicious behaviors such as CC attacks.

The system will record the basic information of all requests to provide data support for subsequent audits. Certification inspection is the core link of the whole security process. The system will extract authentication credentials (such as JWT token or cookie) from the request to verify its validity and authenticity. The verification process includes checking the validity of the signature, validity period and token. For invalid credentials, the system will immediately clear the relevant residual information and redirect the user to the login page. For authenticated users, the system will load their detailed permission context to prepare for subsequent permission control.

After passing the authentication, the system enters the refined authority control stage. This link uses the model of combining RBAC (role-based access control) and ABAC (attribute based access control), which not only checks the user's role permissions, but also verifies the relevant attributes of the request content. The permission system supports permission inheritance at the organizational structure level, and all permission changes can take effect in real time without requiring users to log in again. The system will strictly verify the operation authority of each request to ensure that users can

only access authorized resources.

At the level of data security, the system implements multiple protective measures. All input data will undergo strict disinfection, including removing dangerous HTML tags, filtering special characters, etc., to effectively prevent XSS attacks. For database operations, the system forces the use of parametric queries to completely eliminate the risk of SQL injection. When outputting data, the system will automatically perform encoding conversion to ensure that no malicious script will be executed when the interface is displayed.

5. Conclusion

This paper presents an innovative cognitive enhancement system for data center operation and maintenance (O&M), leveraging heterogeneous computing to address the dual challenges of real-time monitoring under resource constraints and skill development for O&M personnel. By integrating a "monitoring-education" collaborative framework, the system achieves a dual-loop mechanism: a monitoring loop ensures millisecond-level device response through CPU multi-core parallel computing, while a learning loop dynamically updates knowledge graphs and delivers personalized resources via GPU acceleration. The three-tier architecture (presentation-logic-data layers) and hardware acceleration using Intel Core i5 processors and Galaxy GTX graphics cards enable professional-grade O&M capabilities in consumer-grade environments.

References

- [1] Ferrari L, Marsay G, Park C M, et al. Future Directions: Key Considerations for Successful Design and Implementation of SEL Initiatives[M]. Springer, Cham, 2025.
- [2] Zhang Y, Li W, Chen X. A Survey on Real-Time Monitoring Systems for Data Center Operations[J]. IEEE Transactions on Industrial Informatics, 2023, 19(4): 2456-2468.
- [3] Liu S, Wang J, Yang H. Design and Implementation of a Java-Based Monitoring Framework for Heterogeneous Computing Platforms[J]. Journal of Systems Architecture, 2024, 135: 102345.
- [4] Kumar A, Singh P. Enhancing Cognitive Skills Through Adaptive Learning Systems: A Case Study in Technical Education[J]. Computers & Education, 2024, 188: 104567.
- [5] Chen L, Zhou M, Huang R. Performance Optimization of Database Access Layers in Enterprise Applications Using MyBatis-Plus[J]. Software: Practice and Experience, 2023, 53(5): 1123-1145.
- [6] North S, Lafky R, Horgos B, et al. Collaboration in Action: Successful Implementation of a Learner-Driven Virtual Interprofessional Education Curriculum in the Clinical Learning Environment[J]. Journal of Interprofessional Care, 2024, 38(4/6): 38.
- [7] Wang F, Li Z, Zhang K. A Secure and Efficient Access Control Mechanism for Web-Based Monitoring Systems[J]. IEEE Access, 2024, 12: 56789-56801.
- [8] Patel R, Joshi S. Leveraging GPU Acceleration for Real-Time Knowledge Graph Updates in Educational Platforms[C]. 2024 International Conference on Advanced Computing and Communication Systems (ICACCS), 2024: 1-6.
- [9] Brown T, Davis K. Integrating RBAC and ABAC for Enhanced Security in Enterprise Applications[J]. Journal of Information Security and Applications, 2023, 68: 103289.
- [10] Buchanan C J, Young E, Mastalerz K A. Engaging Resident Physicians in the Design, Implementation, and Assessment of Bedside Interdisciplinary Rounds[J]. Journal of Interprofessional Care, 2024, 38(3): 469-475.