Original Research Article

Design of an intelligent stadium management system

Qijia Yu, Shuning Lv

Liaoning University of Science and Technology, Anshan, Liaoning,114051, China

Abstract: This paper aims to design an intelligent stadium management system, which is aimed at the complex needs of university stadium management, and uses modern information technology to realize the intelligent scheduling and management of stadium resources. By integrating the functions of venue booking, course scheduling and membership management, the system can significantly improve the management efficiency and optimize the user experience. Based on the advanced web platform, the system ensures real-time updating and extensive access to information and provides powerful support for the modernized management of university stadiums.

Keywords: Intelligent stadium management system; College stadiums; Web platform; Resource scheduling; Informationized management

1. Preface

With the booming development of education and the increasing awareness of national fitness, the management and operation efficiency of university stadiums, as an important place for physical education, sports training and community sports activities, are directly related to the quality of teaching, student health and the effective use of sports resources. However, the traditional manual management mode often faces the problems of untimely information updating, irrational resource allocation, cumbersome reservation process, etc., which is difficult to meet the growing diversified needs. Therefore, it is particularly important to develop an efficient, convenient and intelligent stadium management system[1-2].

In the next section, we will discuss in depth the design ideas, functional modules, technical realization and practical application effects of the intelligent stadium management system, with a view to providing reference and reference for the development of similar systems[3-4].

2. The design of a Web-based platform for college stadium management system, the

2.1. Topology design

Topology design is one of the key steps to build a Web-based college stadium management system, which defines the connection and interaction logic between system components. The following are the main components of the system topology design.

Client: Users access the system through a Web browser, which is the main interface for users to interact with the system. The client does not need to install additional software, but only needs to have a network connection.

Web server : responsible for receiving client requests , processing business logic , and return the corresponding results.Web servers usually run Web applications , such as Java-based Servlet or Spring Boot applications , as well as Web server software for processing static resources , such as Apache HTTP Server or Nginx.

Application server: In the application logic is more complex, the Web server may be part of the business

logic entrusted to the application server processing. Application servers provide the business components of the operating environment , support for distributed computing and load balancing[5] .

Database server: Stores all the data of the system, including user information, venue booking records, course schedules, etc. The database server runs a database management system (DBMS), such as MySQL, PostgreSQL, etc., to provide data storage, retrieval and management functions [6] .

Backup and recovery system: To ensure data security and availability, the system should be designed with a backup strategy that regularly backs up database data to a secure storage medium. At the same time, a data recovery mechanism should be provided so that data can be quickly recovered in case of loss or damage.

Network security equipment: Includes firewalls, intrusion detection systems (IDS) and virtual private networks (VPNs) to protect systems from external attacks and unauthorized access [7-8] .

2.2. Database design

Database design is the core part of the Web-based college stadium management system, which defines the data storage structure, relationships and constraints in the system. The following are the main steps and contents of the database design of the system.

Requirements analysis: Define what data needs to be stored in the system and the relationship between these data. For example, user information, venue information, booking records, course scheduling, etc. are all key data that the system needs to manage.

Conceptual design: An entity-relationship diagram (ER diagram) is used to describe the data model. In the ER diagram, entities (e.g., users, venues) are represented by rectangles, attributes (e.g., user name, venue name) are represented by ellipses, and relationships (e.g., users booking venues) are represented by diamonds. Through the ER diagram, the association between data can be visualized.

Logical design: Converts the conceptual design into a concrete database table structure. A table is created for each entity, and the columns in the table correspond to the attributes of the entity. At the same time, foreign key relationships between tables are defined to maintain data consistency and integrity[9-10] .

Physical design: determine the database table storage structure, indexing strategy, partitioning scheme. Physical design directly affects the database query performance and storage efficiency.

Security design: set the access rights of the database to ensure that only authorized users can access and modify data. At the same time, the implementation of data encryption and backup strategy to protect the confidentiality and integrity of data.

Optimization and testing: optimize the performance of the database, such as adjusting indexes, optimizing query statements. At the same time, functional testing and performance testing to ensure that the database design meets the system requirements.

Database design should follow the principle of normalization to reduce data redundancy and improve data consistency. At the same time, taking into account the actual needs and performance requirements of the system, it may be necessary to standardize a certain degree of compromise. Reasonable database design can ensure the accuracy, integrity and efficiency of system data.

3. The realization of college stadium management system based on Web platform, the

3.1. Topology realization

The realization of topology is the basis for ensuring the stable operation of the whole management system.

Web-based university stadium management system usually adopts B/S (Browser/Server) architecture, i.e. browser/server mode. Under this architecture, users access the system through a Web browser, while the server is responsible for handling business logic and data storage.

Hardware deployment: Servers are usually deployed in data centers to ensure high availability and security. Servers may include Web servers, application servers, database servers, etc., which are connected via high-speed networks.

Software Configuration: Install the appropriate operating system, web server software (e.g., Apache, Nginx), application server software (e.g., Tomcat, Spring Boot) and database management system (e.g., MySQL) on the server.

Network setup: Configure network firewalls, routers and other network equipment to ensure system network security and access control. At the same time, set up load balancer to improve the concurrent processing capacity of the system.

Security policy: implement HTTPS protocol to ensure the security of data transmission. At the same time, set up user authentication and authorization mechanisms to prevent unauthorized access.

3.2. Implementation of functional modules

A Web-based college stadium management system usually contains the following functional modules.

User Management Module: Realize the functions of user registration, login, password modification and permission management. Users can be divided into administrators, teachers, students and other roles, each role has different rights and functions.

Course Scheduling Module: Teachers can arrange the time and place of the course according to the teaching plan, and students can view the schedule and select courses. Administrators can manage and adjust the course schedule.

Membership Management Module: It realizes the functions of entering, modifying, deleting and inquiring membership information. Members can enjoy specific benefits and services, such as priority booking of venues, free to participate in activities and so on.

Due to space constraints, I can't provide a complete, working implementation of the core code, but I can give you an overview of the core logic and possible code snippets for each functional module. Please note that these code snippets are simplified and assume that you already have a basic web application framework (e.g. Spring Boot) and database (e.g. MySQL) set up.

the user management module

Core logic.

User registration: Receive the information entered by the user, validate it and save it to the database.

User Login: Verify the user name and password entered by the user, and generate a session or token after success.

Password change: Allow users to change their passwords after verifying their identity.

Privilege Management:Assign different privileges according to user roles.

Code snippet (pseudo-code).

java

// UserService.java

@Service

```
public class UserService {
    @Autowired
    private UserRepository userRepository;

    public User register(User user) {
        // validate user input
        // ...
        return userRepository.save(user);
    }

    public String login(String username, String password) {
        User user = userRepository.findByUsernameAndPassword(username, password);
        if (user != null) {
            // generating sessions or tokens
            // ...
            return "sessionToken";
        }
        return null;
    }

    // Other methods...
}
```

the venue booking module

Core logic.

View Venue Usage:Retrieve booking information for a venue from the database.

Reservation: After verifying the identity of the user and the availability of the venue, the reservation information is saved.

Administrator Review:Administrators can view and process booking requests.

Code snippet (pseudo-code).

java

```
// BookingService.java
@Service
public class BookingService {
    @Autowired
    private BookingRepository bookingRepository;
    @Autowired
    private VenueRepository venueRepository;

    public List<Booking> getVenueBookings(int venueId) {
        return bookingRepository.findByVenueId(venueId);
```

```
    }

    public Booking bookVenue(int userId, int venueId, Date startTime, Date endTime) {
        // Authenticating user identity and site availability
        // ...
        Booking booking = new Booking();
        booking.setUserId(userId);
        booking.setVenueId(venueId);
        booking.setStartTime(startTime);
        booking.setEndTime(endTime);
        // ...
        return bookingRepository.save(booking);
    }

    // Administrator review methods...
}
```

course organization module

Core logic.

Teacher scheduling: After verifying the teacher's identity, course information is saved.

Student View Schedule:Retrieve course information from the database.

Administrators manage courses:Administrators can view, edit and delete course information.

Please note that these code snippets are highly simplified and do not include complete error handling, data validation and security measures. In actual development, you will need to implement and test the code in detail according to your specific needs. In addition, you may also need to use front-end technologies (such as HTML, CSS, JavaScript and frameworks such as React, Vue, etc.) to build user interfaces and interact with back-end services.

4. Application testing of the management system

Comprehensive and detailed testing is an indispensable part of a management system before it is put into operation. This step is aimed at verifying that the system meets the predefined functionality and performance requirements, as well as identifying and resolving any bugs that may exist, and is described in more detail in the following application testing of the management system.

4.1. Purpose of the test

Application testing of the management system is aimed at.

Verify Functional Realization: Ensure that the system is capable of performing all designed functions as expected.

Detecting performance: Evaluating the responsiveness, stability and compatibility of the system under high load conditions.

Identify potential problems:Identify possible bugs, vulnerabilities, or deficiencies in the system through testing.

Enhance user experience: Optimize the system based on the test results to improve user convenience and satisfaction.

4.2. Test content

Application testing of the management system consists of two main areas: functional testing and performance testing.

Performance testing is done through black-box testing (also known as behavioral or functional testing), which focuses on the external behavior and performance of the system. Testers will simulate different usage scenarios and load conditions to test the system in order to evaluate its performance level.

Load test: simulate a large number of users accessing the system at the same time, evaluate the system's processing power and response time.

Stress test: Test the system when it reaches or exceeds its designed load, and observe whether the system will crash or performance degradation.

Compatibility testing: Test the system on different browsers and operating systems to ensure that the system works properly and has good compatibility.

4.3. Test summary

After a comprehensive application test, the management system shows excellent performance in terms of function and performance. The system is able to realize all the designed functions stably and meet the demand of automated management system for university gymnasiums; at the same time, the system is able to maintain good performance and compatibility under high load conditions, providing users with a smooth experience of using the system. Therefore, it can be considered that the system has the conditions to be formally put into use.

5. Conclusion

With the successful completion of this project, we not only have a deeper understanding of the development and application of management systems, but also witnessed how technology plays an indispensable role in the actual needs. From the early stage of the project requirements analysis, to system design, development, testing, until the final deployment and application, each step has condensed the team's wisdom and sweat, reflecting our pursuit of technical excellence.

At the same time, we also realize that the development of technology never ends. Although the system has achieved remarkable results in the current stage, but in the face of changing market demand and technology trends in the future, we still need to maintain a keen insight and continuous innovation, and constantly iterative upgrading of the system in order to adapt to new challenges and opportunities.

Funding

Liaoning University of Science and Technology College Student Innovation Training Program Standing 2025 Standing one-year period

References

[1] An Haichao. Design and realization of Web-based science and technology management system[J]. Information Record Material, 2022(06).34.

[2] P. Zhang. Research on the design of Web-based work management system for teachers in private colleges

and universities[J]. Information and Computer (Theoretical Edition), 2017(19).45-47

[3] Li Xiaotong, Zhao Hongkai. Design and development of online student leave signing management system based on Web technology[J]. Laboratory Science, 2021(01).123-128.

[4] Han X. Design and realization of WEB-based news management system[J]. Electronic World, 2020(03).45-47.

[5] ZHAO Dongyuan, JI Xiaobo, ZHANG Jie, LU Ming. Design and realization of Web front-end for court file cabinet management system[J]. Industrial Control Computer, 2022(03).230-234.

[6] Li T. Construction of departmental routine management system based on WEB[J]. Industry and Technology Forum, 2008(06).5.

[7] Xiao Zheng, Wang Shuang, Huang Guofan. Realization of Web query page of seat management system[J]. China Education Informatization, 2013(15).17-18.

[8] Mingsheng Zhang. Web-based work order management system design[J]. China Cable TV, 2019(01).23-24.

[9] YU Qiang, LIAO Xiaobin, LIN Jianbin. Development of a Web-based modern medical device management system[J]. Chinese Journal of Medical Devices, 2017(01).23-24.

[10] Yu Lufei, Zhao Quan. Analysis of WEB-based multimedia material library management system[J]. Electronic World, 2016(22).66-67.