Video Website Management System Based on Database SQL

Yixuan Gao
School of Information and Communication Engineering
Communication University of China
Beijing, China
gyx_0611@163.com

Xinran Ba
School of Information and Communication Engineering
Communication University of China
Beijing, China
* Corresponding author: baxinran@cuc.edu.cn

***Abstract*— With the advance of the Internet, short videos have emerged as the most popular method of social networking. Consequently, individuals now have access to vast amounts of unstructured short video data. Efficiently managing, organizing, and retrieving this large volume of short video data, along with the associated interaction data on the network, has become a highly relevant research topic. Moreover, most short video platforms have lengthy development cycles and require substantial storage space. In light of these issues, this paper develops and implements a practical and meaningful short video website management system based on SQL (Structured Query Language) database. The functional design of the system is divided into two parts: the client-side and the server-side. The client-side implementation employs HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), and JavaScript for page design, resulting in a user-friendly interface with quick startup time, local storage capability, and device compatibility. On the other hand, the server-side utilizes the SpringBoot framework to construct the API (Application Programming Interface) backend and employs MyBatis for seamless interaction with the MySQL database. MySQL, a relational database, is employed to store the server-side data.**

Keywords—Javaweb, MySQL database, short video

I. INTRODUCTION

Today, the Internet is undergoing rapid development, and short video applications have the opportunity to capitalize on the 5G and artificial intelligence era, thereby enhancing economic benefits. By the end of 2022, video content already constituted nearly 80% of global mobile data traffic, and video content has become increasingly refined [1].

The objective of this study is to develop a short video website management system based on an SQL (Structured Query Language) database. The system aims to efficiently organize and manage videos and their associated data, fulfill user requirements, and ensure ease of use and system security. Initially, the functional and non-functional requirements of users were gathered and analyzed. The functional design of the system was divided into two parts: the client-side and the server-side. The client-side implemented various functions such as login and registration, personal information management, video management, interaction, message management, and data visualization. On the other hand, the server-side handled information management and system maintenance. Additionally, a database structure was designed to store the server-side data. Furthermore, the study conducted comprehensive testing of the system's functions and performance, and the test results aligned with the expected outcomes.

II. RELATED TECHNOLOGY

A. JavaWeb Technology

In recent years, Java has been widely used for web development, Android APP development, and big data technologies. Therefore, the development of the server-side of the short video website management system are based on JavaWeb [2]. This system uses SpringBoot framework to build the server-side and MyBatis to interact with database MySQL.

1) The SpringBoot framework, which significantly simplifies the number of configuration files in the SSM (Spring+Spring+MyBatis) framework, is widely used in JavaWeb development. SpringBoot favors convention over configuration, that is, the needs of the convention should be replaced only if the agreement does not meet expectations.

2) MyBatis is an open-source framework for the persistence layer that simplifies the implementation of database access in Java applications. MyBatis input mapping encapsulates the parameters required for SQL operations. Output mapping defines the mapping between SQL execution results and Java types. Mybatis is a good semi-ORM (Object Relational Mapping) framework. Additionally, manual SQL configuration makes it easier to implement the SQL optimization. Therefore, MyBatis is more flexible than fully automated ORM framework [3].

B. Database Technology

A database is a computer software system that manages structured information or data. It serves as a collection of organized data. The core of a database is a file system [4]. Databases can be broadly categorized into two main types: relational databases and non-relational databases. Relational databases encompass systems such as Microsoft Access, Oracle, and MySQL. These databases primarily store data in the form of tables. Relational databases are known for their ease of maintenance, user-friendliness, and support for complex queries. However, they may lack flexibility in certain scenarios. Non-relational databases, on the other hand, are designed to store unstructured data in a structured manner. These databases utilize various structured storage methods such as documents or key-value pairs. Examples of non-relational databases include Redis, MongoDB, and Neo4j. While non-relational databases

2023 International Conference on Culture-Oriented Science and Technology (CoST) | 979-8-3503-5799-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/COST60524.2023.00037

979-8-3503-5799-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CoST60524.2023.00037

offer flexible data formats, they can have more complex data structures. For the short video website management system, MySQL, a relational database, has been chosen due to its small size and open-source nature. It provides efficient data storage and query capabilities for the system.

C. Other Technology

The server-side API (Application Programming Interface) interface of this system is developed using the Swagger2 framework, which enables efficient documentation and testing of the API. On the client-side, the Thymeleaf template engine is employed to separate the interface from the data, facilitating easy testing and modification of the interface. The client-side uses the Bootstrap framework and jQuery library. Bootstrap is based on HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), and JavaScript for developing responsive layout, mobile-first web projects, jQuery, often used in conjunction with Bootstrap, provides compatibility across multiple browsers. For audio and video streaming functionality, the FFmpeg framework is utilized. It offers comprehensive multimedia processing capabilities. As for the web-based video player, Flowplayer is employed.

III. Requirements Analysis

Based on the analysis of the system functions and user demographics, the system roles have been categorized into three types: Guest, Individual User, and System Administrator.

A. Guest and Individual User Requirements Analysis

Guests are required to implement account registration, and they can become individual users after registering an account. Individual users are required to implement login, information management, video management, video publishing, video discovery, video playback, and interaction [5]. Guest and individual user requirements analysis is shown in Fig. 1.

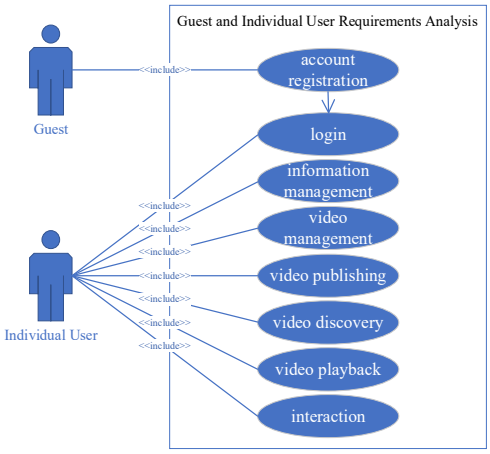

Fig. 1. Guest and Individual User Requirements Analysis.

B. System Administrator Requirements Analysis

The background management system is mainly participated by the system administrator to maintain the relevant resources of the client-side. The main requirements are the management of information and the statistical analysis of relevant data to visualize relevant data. Information management mainly includes user management, video management, role management, and tag management [6]. System administrator requirements analysis is shown in Fig. 2.

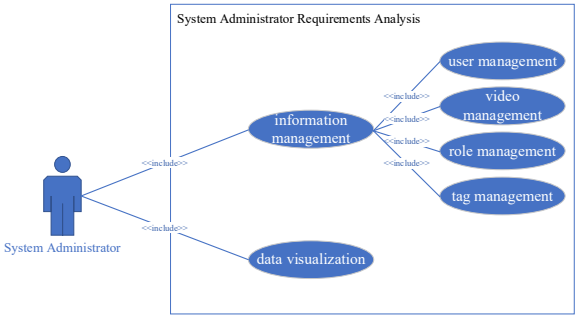

Fig. 2. System Administrator Requirements Analysis.

C. Non-functional Requirements Analysis

Non-functional requirements specify the quality attributes of the software and define the general characteristics of the system, behavior. The following points will analyze the requirements in terms of: (1) security, (2) availability, and (3) performance.

1) The system stores data in the relational database MySQL. Only the root account has all the permissions of the database. Others can only get permissions granted by root. For MySQL, there are encryption fields and permission fields in the system tables. Additionally, to prevent data loss, the database can be backed up. The configuration environment is strictly monitored on the server-side. Moreover, to ensure system security, the permission restrictions of different users are performed on the client-side.

2) The availability of the system implies that the user can complete the individual operations quickly and easily. Therefore the interface is designed to be as simple as possible, so as to improve the efficiency of users and system administrators. For the database, the establishment of tables should be easy to understand and maintain.

3) The system should be capable of supporting the storage and access of a large number of video data. It needs to strengthen the response performance and improve the efficiency of data processing to let users interact with the system better [7].

IV. Short Video Management System Design

A. Development Tools and Operating Environment

The operating system is Windows 11. The server-side is developed using Intellij IDEA 2022.3.2. The database is MySQL 8.0. The database application tool is Navicat Premium 16. The system development requires the Java 8 environment. The server-side needs to configure Maven 3.6.0, JDK 1.8.0, Tomcat 9.0.17, SpringBoot 2.1.4, Swagger 2.6.1, MyBatis 3.5.11, and FFmpeg 6.0. The client-side needs to configure Thymeleaf 3.0.11, BootStrap 3.3.7, and jQuery 1.12.4. It also uses Intellij IDEA 2022.3.2 to develop, use Microsoft Edge browser to view page design effects.

B. Database Design

The requirements are specified into a conceptual model, which is represented by an E-R (Entity Relationship) diagram. The E-R diagram of this system is shown in Fig. 3. The conceptual structure can be transformed into a data model. Storage structure and access methods that are most suitable for the logical data model and application environment need to be chosen. Finally, add data, develop and debug the application program.

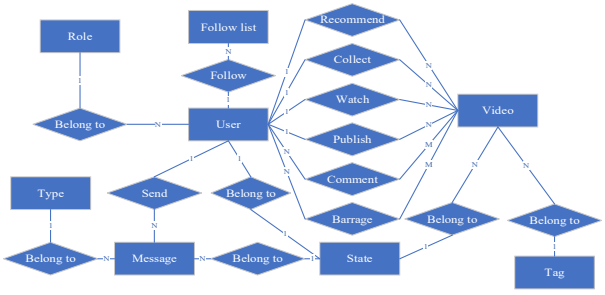

Fig. 3. The E-R Diagram of the System.

This database establishes 12 tables according to the E-R diagram. To introduce the data table, user table, state table, and recommended prediction value table are as examples.

1) User Table

Role_id and state_id are foreign keys, which connect to the role table and state table. Table details are shown in TABLE I.

TABLE I. USER TABLE

Column	Data Type	Constraint
user_id	int	Primary Key, Not Null, Auto Incremental
user_name	varchar	Primary Key, Not Null
user_age	int	/
user_sex	varchar	/
user_mail	varchar	/
user_phone	varchar	/
user_address	varchar	/
password	varchar	/
role_id	int	Foreign Key, Not Null
register_date	datetime	/
fan_num	int	/
icon_url	varchar	/
state_id	int	Foreign Key, Not Null

2) State Table

The state table is a subordinate table of user table, video table, and message table. The states of user information include pending approval, approved, and not approved. The states of video include listed, listing, and removed. The states of message include read and unread. The table details are shown in TABLE II.

TABLE II. STATE TABLE

Column	Data Type	Constraint
state_id	int	Primary Key, Not Null, Auto Incremental
state_name	varchar	/

3) Recommended Prediction Value Table

The predicted value, Pre_star, is calculated using SVD (Singular Value Decomposition) algorithm, which makes recommendations based on the comment stars in the comment table. Comments are rated on the basis of stars ratings, with options ranging from one to five. First, users and their ratings of the videos are put in a matrix. The videos that are not rated by the users are represented as blanks in the matrix. Then, the unknowns are filled by predicted values. The rating matrix is represented by the multiplication of two matrices. They are filled with initial values using the random function. Furthermore, they are iterated using the ALS (Alternating Least Squares) algorithm. Finally two matrices are multiplied. The predicted values are sorted from high to low. Table details are shown in TABLE III.

TABLE III. RECOMMENDED PREDICTION VALUE TABLE

Column	Data Type	Constraint
pre_id	int	Primary Key, Not Null, Auto Incremental
user_id	int	Foreign Key, Not Null
video_id	int	Foreign Key, Not Null
pre_star	float	/

C. System Functional Design

The SpringBoot framework consists of five main layers: view layer, controller layer, service layer, mapper layer and model layer. The view layer interacts directly with users who perform operations. Furthermore, the browser sends http protocol requests, which are received by the Tomcat server started during the project startup process. The server encapsulates the requests into request objects. The controller layer handles these operation requests or data transmission requests. If business processing is required, it will call the service layer. Controllers in the controller layer transfer data to the service layer and call the interface of the service layer. The service layer performs the logical processing of the business. If it needs data addition, deletion, modification and checking, it will call the mapper layer. The mapper layer is to add, delete, modify, check and perform other operations for the database. Mybatis is on the role of this layer. Moreover, the specific implementation code is in the mapper.xml. Finally, the model layer stores entity classes, and data tables correspond to.

1) Login and Registration Page

The login and registration page includes two functions: user registration and user login.

Two input fields are used to type in the username and password on the registration page. Furthermore, clicking on the registration button can complete the registration. There is a login button below. Clicking it can switch the login page. The registration information that is entered cannot be empty. The login page will be switched after registration is completed.

The login page is similar to the registration page. After logging in, the users can enter the home page. If the login fails, a warning will appear.

2) Home Page

The home page mainly includes the functions of video discovery and video search. The video discovery function mainly includes searching the videos with tags and popular list recommendations. The video search box is on the top right

corner of the home page. The input box allows users to enter the keywords for search. The popular list shows the top 10 videos with the most likes. The bottom of the list shows the rotation of popular videos. The home page is shown in Fig. 4. Additionally, the home page lists the tag buttons, click on them to jump to the corresponding tag position. Videos with tags is shown in Fig. 5. There are several options at the top of the home page to switch pages. Users can manage personal information and account information on the personal center page. In addition, users can send moments and manage messages. Moreover, users can manage self-published videos on the video center page.

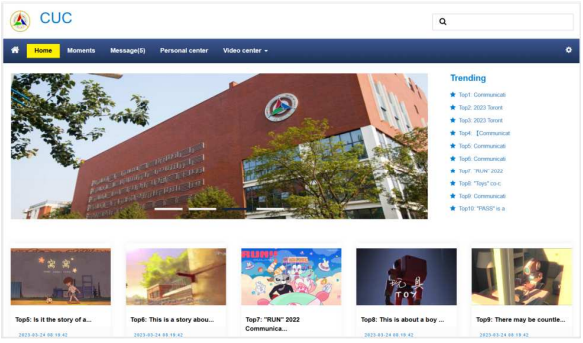

Fig. 4. Home Page.

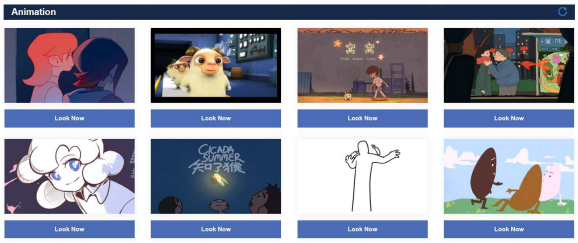

Fig. 5. Videos with Tags on the Home Page.

3) Video Playback Page

The center of the page shows a video player. Users can click on the video to pause or play. The above portion shows the author's profile picture and username. Others can follow and message the author. The right side lists the information about the video. Users can like, comment, and collect the video. Additionally, users can send bullet chats. Moreover, this page recommends other videos at the bottom. Recommended videos are decided by predicted values in the recommended prediction value table. The video playback page is shown in Fig. 6.

Fig. 6. Video Playback Page.

4) Video Publishing Page

The video publishing page is shown in Fig. 7.

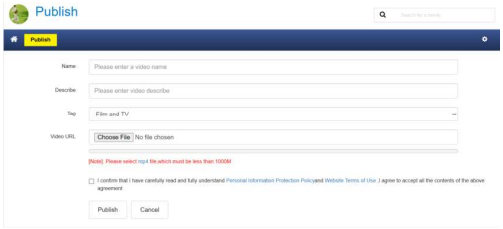

Fig. 7. Video Publishing Page.

Users are required to fill in the video information and select the video file in turn on the video publishing page. The video can be uploaded, and "upload successful" message will be shown when the progress bar is completed.

5) Back-end Management System

The back-end management system mainly includes information management and data visualization functions. Information management includes user management, video management, role management, and tag management. The following is an example of video management. The video management function includes video review and video deletion, the state of videos will be changed from pending approval to listing after review. The video management page is shown in Fig. 8.

The data visualization function counts the total number of views, the number of users, the number of likes that each user gets, the number of views of each tag, etc. The system shows them in the form of pie charts and bar charts. Compared to other short video management systems, this is the advantage of the back-end management system. The data visualization page is shown in Fig. 9. and Fig. 10.

Fig. 8. Video Management Page.

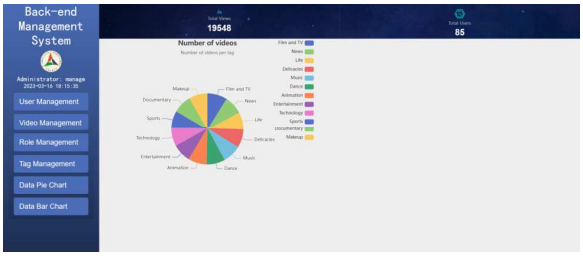

Fig. 9. Visualization of the Number of Videos Per Tag.

Fig. 10. Visualization of Video Views and Likes.

D. *Functional and Performance Evaluation*

For the functional test, the login and registration functions can make judgments regarding the roles. The results are consistent with the expectations. For the client-side, user information management, video playback, video publishing, video discovery, video search, and interactive functions are all implemented. Furthermore, the results of each function are consistent with expectations. The system administrator can add, delete, and check for user, video, role, and tag information. The data visualization function can see the total views, total users, views of videos with each tag, etc., which are presented in the form of bar charts and pie charts. The information management function and data visualization function are consistent with expectations. The server-side is capable of performing information management and system maintenance by operating on the program code, which is consistent with the expectation.

For the performance test, about 500 videos with different tags are uploaded in the system during the test. Furthermore, the time consumption, response time, and CPU occupancy at system startup were monitored. The expected performance indicators for the first run of the system were less than 5 s, which is less than 100 ms for the response time of switching pages, and less than 10% CPU occupancy. The results of the performance evaluation were consistent with the expected performance indicators.

V. CONCLUSION

The system was designed in detail based on the requirements analysis. The design of the system was divided into two parts: the client-side and the server-side. For the client-side implementation, HTML, CSS, and JavaScript were utilized for page design. This combination allowed for the creation of visually appealing and user-friendly views and also focused on ensuring a short start-up time, leveraging local storage capabilities, and ensuring compatibility across different devices. The server-side builds API backend using SpringBoot framework. It uses MyBatis framework to interact with MySQL. It can achieve efficient management of the relevant data. An advantage of this system over other short video management systems is the data visualization. This function allows the administrator to grasp and manage the client-side data more efficiently. Compared to other short video management systems, this is the advantage and innovation of the back-end management system. Furthermore, the administrator can clearly deduce whether the system needs maintenance and other information through these data. This system achieves the combination of new media and short video, which enriches the way of information dissemination. It meets the user needs and ensures the ease of use and security of the system.

ACKNOWLEDGMENT

This work supported by “the Fundamental Research Funds for the Central Universities” under Grant CUC230B050

REFERENCES

[1] Z. Lei, “Design and Implementation of Short Video Production and Sharing Platform Based on Android,” Beijing Jiaotong University, 2021.

[2] L. Xie, “Design and Implementation of an Integrated Platform for Multi-language Methods Based on Javaweb,” 2020 4th Annual International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, pp. 360-363, 2020.

[3] X. Shiyong, D. Tianxiang, Z. Rongzheng and W. Rongsen, “Research On Mybatis Mapper Model Based On SQL Template,” 2020 International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China, pp. 502-505, 2020.

[4] C. Mariani, M. Mercês, M. Holanda, E. D. O. Ribeiro and M. C. Victorino, “Hive and MySQL - Query performance comparison for a multidimensional model,” 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, pp. 1-6, 2022.

[5] L. Xing, L. Zhicong, “Fifteen Seconds of Fame: A Qualitative Study of Douyin, A Short Video Sharing Mobile Application in China,” Meiselwitz, G. Social Computing and Social Media. Design, Human Behavior and Analytics. HCII 2019. vol. 11578, pp. 233–244, 2019.

[6] Y. Guo, “Design and Implementation of Travel Website Based on Java Web,” 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, pp. 659-663, 2023.

[7] M. Chen, T. Liu, W. Yu and J. Gao, “Online Florist System Based on JavaWeb,” 2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Hangzhou, China, pp. 570-574, 2021.