

Research Article

Development of Music Online Teaching System Based on Distributed Database Search Technology

Wang Shun 

Conservatory of Music, Guangxi Normal University, Guilin, Guangxi 541006, China

Correspondence should be addressed to Wang Shun; 2007052@muc.edu.cn

Received 27 June 2022; Revised 12 July 2022; Accepted 22 July 2022; Published 29 August 2022

Academic Editor: Imran Shafique Ansari

Copyright © 2022 Wang Shun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous advancement of Internet technology, the traditional music teaching model has slowly evolved from offline to online. Nowadays, music platforms no longer focus solely on music but are supplemented by some social functions. Under the current huge market demand, in order to meet the individual needs of different music users, more and more music platforms continue to perform technological updates. Various music platforms have sprung up continuously. Although they provide users with a variety of options, at the same time, users must download corresponding applications to experience each music platform, which brings a certain degree of confusion to users. In order to solve such problems, this article implements a download-free music online education system through PWA. Aiming at the shortcomings of traditional database search engine systems, this article studies the implementation methods of distributed database search engines. First, a search engine architecture based on a cloud database is proposed so that the existing database search engine system can be implemented in the HBase distributed database, thereby improving the efficiency of the search engine. Secondly, a new data preprocessing method is proposed, which improves the data interaction technology between the relational database and the HBase distributed database, and simplifies the project development framework. Subsequently, this paper proposes a multiuser detection algorithm, which can determine duplicate data entries by calculating keyword similarity to improve the accuracy of search results. This paper promotes the development of music teaching by integrating a multiuser detection algorithm and distributed database search technology and applying it to the construction of music teaching online system.

1. Introduction

An online music teaching system is an effective supplement to traditional learning methods. The system can provide an efficient, simple, and flexible teaching mode [1]. Through detailed research and analysis of the system, this article introduces it into the online learning mode and completes the learning effect test. The system can complete the teacher-student integrated remote management teaching Mode, so as to improve the level of distance learning ability in all aspects [2]. Combined with the current music teaching needs, this paper selects unified modeling language (UML) to analyze the relevant business process of music teaching [3]. Based on the analysis results, use case diagrams to define basic information management, music theme management, music practice management, online classroom management,

and notification information management are five models, and the nonfunctional system requirements are analyzed and designed at the same time [4]. Based on the JavaEE framework, the article completes the design of multiuser detection algorithm, general system architecture, detailed module structure design, etc., and proposes a multi-layer construction framework, including user visual interface layer, business logic layer, and database layer through UML sequence and class [5]. The figure completes the detailed design plan of functional modules such as independent learning management module, learning resource module, communication and cooperation module, and monthly performance system management module, and then uses vision to carry out the detailed design of the system database model, and conducts the core physical data model plan design, based on this design. Through the Java language and

the JavaEE development platform, this paper completes the technical implementation of the basic information management, music theme management, music practice management, online classroom management, and information notification management modules [6]. The online education system uses a mobile learning system to provide students with a convenient and fast learning platform. Compared with the existing music instruction system, the Android-based music teaching system uses wireless Internet technology, which is more suitable for the fragmented learning habits of today's students, can effectively improve the students' learning ability, and help stimulate their interest in learning [7]. Among them, the system database is very large, so it needs a specific database to run smoothly. Therefore, this article conducts detailed research on the realization of distributed database search engines [8]. By improving the data interaction technology between the relational database and the HBase database, a search scoring mechanism compatible with the Levenshtein distance vector algorithm is proposed, which improves the accuracy of the search results and realizes the system function and performance test [9]. The test results show that the function of the system successfully met the requirements of the expected design goals. The test of building the system under the LoadRunner platform shows that even if 200 users access the system at the same time, it is still stable and able to meet the requirements of the program.

2. Related Work

The literature uses two more mature framework schemes, namely .Net and JavaEE. Most of the system is based on the B/S structure, characterized by strong system scalability, no customer installation, strong user scalability, and the ability to allow new students to join continuously [10]. Due to the development of HTML5, the client can complete more and more tasks, so it is more suitable for online learning systems [11]. The system is suitable for traditional Windows systems in terms of design and functional layout and can be completed in a browser. In addition, because the online learning system often exchanges data with other systems, such as the campus courseware management system, the system must use Web services to realize multisystem data sharing [12]. The literature uses a free MySQL database, and the overall operation of the system is divided into a student front-end learning system and a back-end system administrator or teacher responsible for completing the maintenance system of course resources. The code used to control the playback of the course source is generally located on the front end, and the data are read on the server side; while the background system generally provides the update and maintenance of learning resources, which can be imported in batches and classified according to the table [13]. The literature studies the characteristics of an online learning system based on traditional mobile Internet technology to build an Android music teaching system with new Internet technology, which has a centralized learning exchange platform for teachers and students [14]. Some available functional models are proposed, such as basic information management, music student homework

management, music practice management, online classroom management, and information notification management. The literature proposes a database search engine architecture based on a cloud platform, which enables the database search engine system to be implemented in the HDFS distributed file system. This paper selects the HBase distributed database to achieve this goal and improve indexing and search efficiency [15]. A new method of data preprocessing is proposed, using distributed data interaction tools Sqoop and iBATIS, Spring, and Struts to construct a composite framework to read and write related databases and read and write serial distributed HBase databases through module-specific programs to simplify the project progress framework.

3. Multiuser Detection Algorithm and Distributed Database Search Technology

3.1. Multiuser Detection Algorithm. This section considers an uplink system based on unlicensed NOMA. The NOMA technology adopted by the system is LDS-OFDM, which modulates a small number of K user information, and then transmits it to N subcarriers through a spreading sequence for overlapping transmission. Therefore, within a certain period of time, the superimposed signal received by each subcarrier at the base station can be expressed as:

$$y_n = \sum_{k=1}^K g_{kn} s_{kn} x_k + v_n, n = 1, 2, \dots, N. \quad (1)$$

Combining the received signals on all iv subcarriers, the received signal vector can be obtained as follows:

$$y = Hx + v. \quad (2)$$

In a real communication situation, even if the user is not kept active or inactive during the entire frame, it will continue to send data in adjacent time intervals, and the possibility is very high. This phenomenon is called a temporary correlation between active user groups in adjacent time intervals. The signals received in j adjacent time intervals can be expressed as:

$$y^{[j]} = H^{[j]} x^{[j]} + v^{[j]}, j = 1, 2, \dots, J. \quad (3)$$

In addition, this section defines the support set (active user set) in the vector as:

$$I^{[j]} = \{k: k \in \{1, 2, \dots, K\}, x_k^{[j]} \neq 0\}. \quad (4)$$

The calculation expression to obtain the final activated user set is as follows:

$$P^{[j(i)]} = \text{Max} \left(\left| \left(H^{[j]} \right)^H r^{[j(i-1)]} \right|, \ell \right). \quad (5)$$

Subsequently, in order to correct the miscalculation of the last iteration, the idea of backtracking was introduced, and $I^{(i-1)}$ and $P^{[j(i)]}$ were combined to obtain the user's candidate activation set $c^{[j(i)]}$, using the formula expressed as:

$$C^{[j(i)]} = I^{[j(i-1)]} \cup P^{[j(i)]}. \quad (6)$$

Finally, according to the activated user set $c^{[j]^{(i)}}$, the least squares (LS) algorithm is used to calculate the candidate recovery signals, and the indexes of the ℓ users with the largest absolute value are selected from them to form the $i^{[j]}$ iteration data set. If the user set 0, then its calculation expression is as follows:

$$I^{[j]} = \text{Max}\left(\left|(H_{c^{[j]^{(i)}}}\right)^\dagger y^{[j]}\right|, \ell\right). \quad (7)$$

Calculate the residual signal: after obtaining $i^{[j]}$, the corresponding residual signal can be calculated by the following expression as:

$$r^{[j]} = y^{[j]} - H_{j^{[i]}}^{[j]} \left(H_{j^{[i]}}^{[j]}\right)^\dagger y^{[j]}. \quad (8)$$

Obtain the recovered signal: use the LS algorithm to recover the sparse signal vector sent in the j th time slot based on $i^{[j]}$:

$$\hat{x}^{[j]} = \left(H_{j^{[i]}}^{[j]}\right)^\dagger y^{[j]}. \quad (9)$$

In the $r^{[j]}$ th slot, the energy can be expressed as:

$$\|r^{[j]}\|_2^2 = \left(y^{[j]} - H_{[j]}^{[j]} x^{[j]}\right)^H \left(y^{[j]} - H_{[j]}^{[j]} x^{[j]}\right). \quad (10)$$

Assuming that the sparse signal $x^{[j]}$ sent in the j th time slot is completely and correctly restored, we can get:

$$\begin{aligned} r^{[j]2} &= \left(y^{[j]} - H_{[j]}^{[j]} x^{[j]}\right)^H \left(y^{[j]} - H_{[j]}^{[j]} x^{[j]}\right), \\ \|r^{[j]}\|_2^2 &= \|y^{[j]}\|_2^2. \end{aligned} \quad (11)$$

Then, the average energy of $r^{[j]}$ can be obtained by the following calculation:

$$E\left\{\|r^{[j]}\|_2^2\right\} = N\sigma^2. \quad (12)$$

Therefore, the stopping conditions of the iterative process are as follows:

$$\|r^{[j]}\|_2^2 \leq N\sigma^2. \quad (13)$$

As shown in Figure 1, in the consideration of the license-free simulation system based on NOMA, if the actual number of active users S is 30, the BER performance curve of each algorithm will change with the change of the signal-to-noise ratio (SNR). The simulation results show that in the general signal-to-noise ratio range, the BER of the DACS algorithm proposed in this paper is always smaller than that of the traditional CS-based algorithm, and if the SNR is to be improved, the BER gap between the two algorithms will continue to expand. Compared with the traditional MUD algorithm, the advantage of the DACS algorithm is that it utilizes the temporary correlation between the active sets of adjacent time slots of the user, and uses the backtracking idea when estimating the active user set to improve the estimation accuracy.

However, the simulation results also show that if the BER is equal to 10⁻⁵, the DACS algorithm proposed in this paper

has a loss of SNR output of about 4 dB compared with the DCS-based algorithm. This performance loss is the cost of DACS algorithm to obtain MUD on the premise of a sparse user activity rate because there is always a specific error in estimating the sparseness of user activity. However, if the performance loss is not considered, the DACS algorithm proposed in this paper is very suitable for real mobile communication conditions, because, in a real mobile communication system, it is difficult for a base station to determine the exact activity sparsity of a known user.

Figure 2 shows the BER performance curves of different MUD algorithms. Because in an unlicensed uplink system based on NOMA, if the SNR is equal to 2 dB, the actual number of active users S will change. It can be seen from the numerical curve that the BER of all schemes increases as S increases. This is because, in the MUD algorithm based on CS sparse signal, lower user sparsity will affect the accuracy of sparse signal transmission. However, while S increases from 20 to 100, the BER of the DACS algorithm proposed in this paper is lower than the BER of the traditional CS-based algorithm and the DCS-based algorithm. This performance advantage shows that, compared with the other two algorithms, the detection performance of the DACS algorithm is less affected by the sparsity of user activity.

3.2. Distributed Database Search Technology. The file system provided by HDFS is an open source implementation of GFS (Google File System) on the Hadoop platform. The architecture of the HDFS system is a typical master/slave architecture, with a NameNode node and multiple data nodes, which can provide an interface for accessing applications. NameNode is the management node of the entire HDFS distributed file system. It is mainly responsible for managing and maintaining the HDFS distributed space on the file system, as well as controlling the operation of client files and managing and assigning specific storage tasks. DataNode provides storage and computing services for real file data. The architecture of the HDFS distributed file system is shown in Figure 3.

The distributed file system provided by HDFS has two independent data sets, one is file data and the other is metadata.

- (1) File data refer to the specific content of files stored by users on the HDFS distributed file system. The HDFS partition file system divides file data into blocks according to a specified size and stores them in different DataNode nodes. Backing up data from a file, generally, there are 3 DataNodes, which can effectively prevent a single DataNode from going down, resulting in irreversible damage to the data in the file, resulting in immeasurable losses, thereby ensuring the safety of the data in the file.
- (2) Metadata refers to the information in the file management system in the allocated HDFS file system. The NameNode handles and maintains all metadata. To start the NameNode, the metadata is first loaded into the memory and then combined

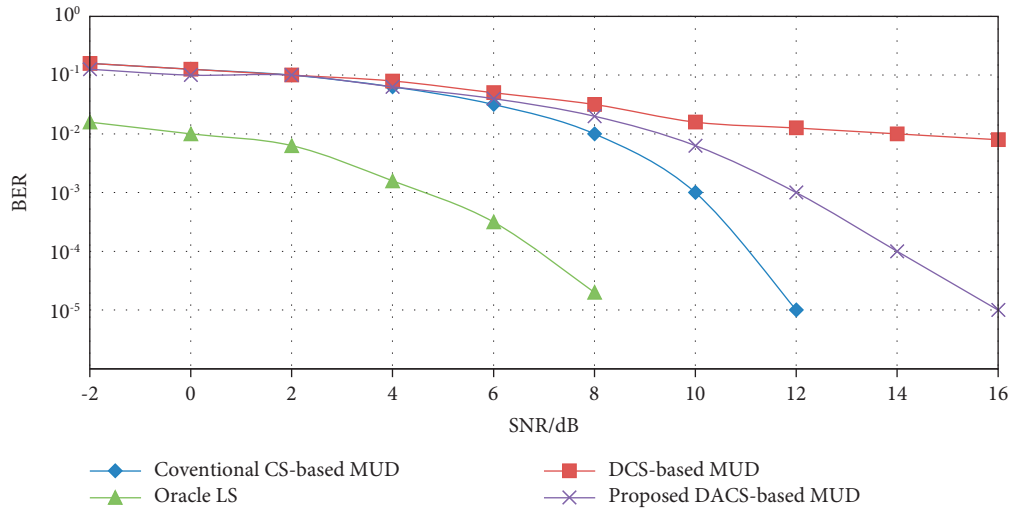


FIGURE 1: BER performance vs. SNR curve.

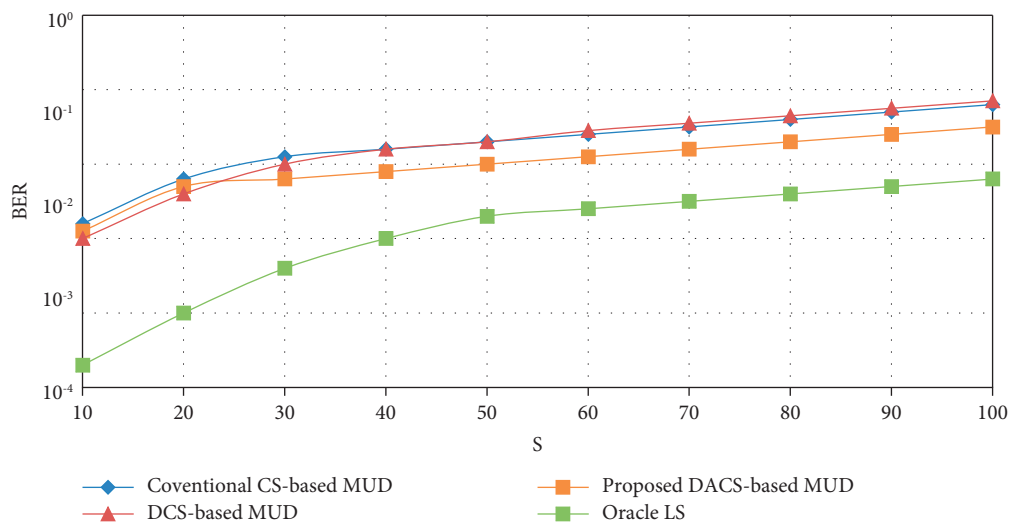


FIGURE 2: BER performance vs. number of active users.

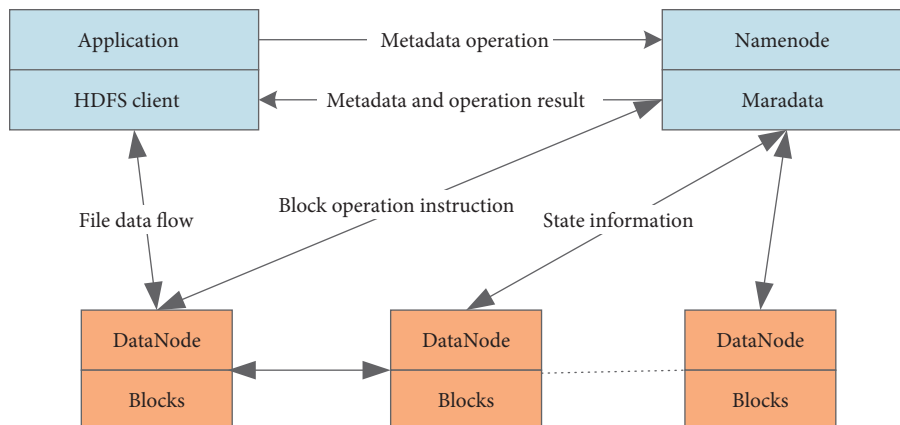


FIGURE 3: HDFS distributed file system structure diagram.

with the metadata information transmitted by the DataNode to form complete metadata. Since the NameNode node in the traditional HDFS distributed file system is a server, the backup of the node is not considered, so if the NameNode node goes down, the HDFS distributed file system will not work normally. In order to meet application requirements, this article supports NameNode node backup when deploying a database search engine system based on a cloud platform, which can effectively solve problems that may occur in the service.

When searching for matching data, a value is often used to define the data to indicate the accuracy of the match and the corresponding result. Determine according to the keyword separation rules, and then use the index method to calculate the similarity of the two keywords. Set the number according to the actual needs of the user so that the result is more accurate. Therefore, this article improves the search algorithm and uses the PageRank algorithm to add the PageRank value of the page and the reference of the page link to the PageRank value of the page assigned to the link to obtain the desired result, as shown in formula:

$$PR(A) = (1 - d) + d \left(\frac{PR(t1)}{c(t1)} + \dots + \frac{PR(tn)}{c(tn)} \right). \quad (14)$$

Among them, PR refers to the PageRank value, $t1 - tn$ refers to page link A, C refers to the number of started web links, and d is the damping coefficient. Experiments show that this method improves the accuracy of search results, but the research content of this article does not include web pages, and the main search is database relational data, so this method is not applicable. Related literature has described the TF-IDF markup technique to evaluate the importance of specific keywords in a document. The importance of keywords is directly proportional to the number of events in the document and inversely proportional to the frequency of occurrence in the document. This method is derived from Lucene's internal marking mechanism. During indexing, each keyword is assigned a fixed value. The score is used to calculate the similarity between the field index and the related field. Lucene provides a formula for calculating the score, as shown in formula :

$$\sum_{tinq} (tf(t \text{ in } d) \times idf(t)^2 \times boost(t, \text{field in } d) \times lengthNorm(t, \text{field in } d) \times coord(q, d) \times queryNorm(q)). \quad (15)$$

The number of occurrences of a keyword in a specified document is the frequency of occurrence of the word. In fact, the words of common keywords in short documents are always lower than those in long documents. Therefore, in order to prevent keywords from being biased toward long documents, whether they are important or not need to be normalized, as shown in formula :

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \quad (16)$$

The document reverse frequency is completely different from the word frequency. It starts with a measure of the pure relevance of documents and the overall importance of keywords. The reverse frequency of a keyword document can be obtained by dividing the total number of documents by the number of documents in the term, and then taking the logarithm of the quotient, as shown in :

$$idf_i = \log \frac{|D|}{|\{d: d \sum t_i\}}. \quad (17)$$

Among them, the numerator represents the total number of documents, and the denominator represents the number of times the keyword appears in the document. The final TF-IDF is shown in formula :

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i. \quad (18)$$

The high frequency of document keywords and the short frequency of document keywords in the entire document will produce a high weight TF-IDF. Therefore, TF-IDF tends

to filter out common words, keep important words, and highlight the relevance of certain words in the document.

In the D-WISE (Distribute-Web Index and Search Engine) algorithm, each database is represented as the document number (Document Freq) corresponding to each object part of the database and the total number of documents in the database. At the same time, all the characteristics of the database properties are described with a detailed internal summary.

$$CV_{ij} = \frac{(df_{ij}/n_i)}{(df_{ij}/n_i) + (\sum_{k \neq i}^N df_{ki} / \sum_{k \neq i}^N n_k)}. \quad (19)$$

The number of member search engines in the metasearch engine system is N , and the proportion of the keyword t_j in the i th database D_i is represented by CV_{ij} . Use CVV_j to represent the CV_{ij} clue validity variance of the keyword t_j , and the calculation formula is as follows:

$$CVV_j = \frac{\sum_{i=1}^n (CV_{ij} - ACV_j)^2}{N}. \quad (20)$$

GLOSS algorithm: set a critical value for each query. If the similarity of the queried object exceeds the critical value, the document is classified as a document of interest. The total equivalence of all documents whose equivalent value exceeds the established standard in the current query is defined as being related to the member search engine database of the given query. Therefore, the score in the current database is the search engine correlation value of the

member, and the estimated value of the correlation degree in the current database is as follows:

$$\text{usefulness}(D, q, T) = \sum_{j=1}^p q_j \times w_j + df_p \times \sum_{j=p+1}^k q_j \times \frac{w_j}{df_i}. \quad (21)$$

Count the number of useful documents in the member search engine database as:

$$\text{NumDoc}(D, q, T) = \text{card}(\{d \mid d \in D \wedge \text{sim}(q, d) > T\}). \quad (22)$$

Generated by the function as:

$$(p_1 \times X^{w_1 \times q_1} + (1 - p_1)) \times (p_2 \times X^{w_2 \times q_2} + (1 - p_2)) \times \dots \times (p_k \times X^{w_k \times q_k} + (1 - p_k)). \quad (23)$$

After expanding and merging the same items, you can get:

$$a_1 \times X^{b_1} + a_2 \times X^{b_2} + \dots + a_c \times X^{b_c}. \quad (24)$$

4. Research on the Design and Implementation of Online Music Education System

4.1. System Requirement Analysis. Demand analysis is based on the user's vision, thinking about the user's demand for the product, and then designing a model suitable for the entire system according to the user's demand. As an online music platform, users should have the following basic requirements for the platform:

- (1) It can be connected and registered, especially on third-party connection interfaces such as QQ and WeChat.
- (2) Play the song, play and display the song information.
- (3) To search for songs, users can search based on keywords such as artist name, song name, album name, and lyrics.
- (4) Song reviews, that is, song reviews are centered on the song.
- (5) News notifications, allowing users to obtain real-time news subscriptions, etc.
- (6) Desktop, which satisfies the user's habit of operating and using mobile applications.

The system needs to use activity charts and use case details to better describe the platform for users to use and understand. The performance that a system can reflect is not a problem that users need to care about, but a well-performing platform can improve user experience and increase user stickiness. Therefore, as a developer, you need to ensure that users not only experience the services they see but also ensure the service experience they cannot see. According to Amazon statistics, if the loading speed of the website page is within 2000 ms, it is the best time for user experience. After more than 2000 ms, every additional 1s will reduce the popularity of the website by 10%. Since the online music

management system runs in the browser, the creation of a music evaluation platform website requires the website to have good performance. There are three points to check when creating a website, such as white screen time, first screen time, and total download time.

White screen time refers to the time from when the user opens the page to when some content is displayed on the page. It is calculated as the time when the page starts to play and render, that is, the time of the first byte and the time of completing the HTML download plus the time of loading the resources in the header. During the blank screen time, the user can neither see anything nor do anything. Therefore, the shorter the white screen time, the better. Usually, the current method is to load some predisplayed HTML files first and then load the JS files.

First screen time refers to the time required for the user's browser to display all content on the first screen. This item is usually an important user experience factor for a website. If the first screen time is less than 5 s, it is better, if it is less than 10 s, it is acceptable, and if it exceeds 10 s, it is unbearable. If the first screen time exceeds 10 s, users will often choose to refresh the page or leave immediately. The total download time is the page load time and the time required to load and display all page sources.

4.2. System Module Design. The system function module diagram is shown in Figure 4:

User management module: The user management module includes registration, login, and download management. The registration authentication module provides users with one registration method and three authentication methods. The registration method is to register with a mobile phone number, which can effectively associate the user account information with the mobile phone number. It can be used to easily retrieve the password if the password is forgotten, and there is a confirmation code for confirmation every time you log in elsewhere. Ensure the security of users' personal information. There are three ways to log in: mobile phone number, WeChat, and QQ.

Search module: The search module provides search services, allowing users to get the song information they want the first time. The system provides users with keywords such as song name, artist name, album name, lyrics, and other information to search, and the search results are displayed in a streamlined manner. **Message notification module:** Push notifications to provide users with the most real-time message notifications. When the system has a message notification, the user can receive the server's message as soon as possible. If the user is currently offline, the notification message can be saved temporarily, and if the user is online, they will be notified of the specific message information.

Song playback module: The playback module is divided into four modules: song playback, song display, song comment, and local song search. The song is played by the player and controlled by the user. For example, fast forward, fast reverse, fixed-point playback, stop, start, and other melody operations. Display songs to show users information

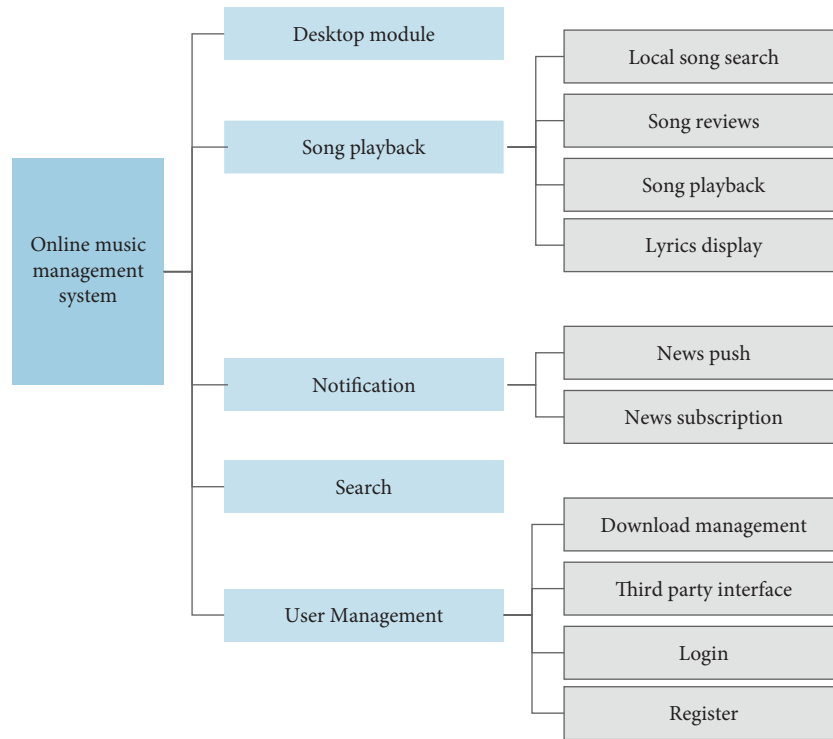


FIGURE 4: System function module diagram.

about the song, such as lyrics and singers. At the same time, the display of lyrics is accompanied by the playback of the song. If the song is played in a specific version, the real-time lyrics will appear in the middle of the lyrics display, and the lyrics at other times will be marked with bright colors. Song comments perceive social needs, and users can communicate with other users in the song comment area. To locate local songs is to add or scan the user’s storage space by customizing the user to add music files downloaded by the user to the player for playback.

Desktop module: The desktop module can be used by the user to add the entire web page to the desktop of the mobile device and provide desktop icons for the user to view. If the user clicks on the desktop icon, a startup animation will appear. After the animation is completed, the interface is similar to the application effect. The web page URL and navigation bar are not displayed, and the entire web page occupies the entire screen.

This article designs the database table according to the conceptual structure of user entity, singer entity, song entity, album entity, and other entities. The following is the detailed introduction of the user table, song table, album detailed information table, comment table, reply table, and other related key databases. The user table is shown in Table 1:

The user table is a table that stores the user’s personal information, and the user_id field is used as the primary key of the table. In addition to the user’s registration information on this platform, the user table has two fields weixin_id and qq_id to store the WeChat id and QQ id of the user, who logs in through the third-party interface.

The song table (Table 2) is a database table used to store song-related information, in which song_id is the unique identifier of the song, and album_id and singer_id are the foreign keys of the song table, which are, respectively, associated with the album table and the singer table.

The album table (Table 3) stores album information, where album_id is a unique identifier, and singer_id is associated with the singer table as a foreign key in the album table.

The comment table (Table 4) is a table that stores user comments on songs, where comment_id is a unique identifier of the table, and song_id is a foreign key associated with the song table, indicating the song under which the comment is a comment. user_id is a foreign key associated with the user table, indicating the id of the user, who posted the comment to_user_id indicates that the @ function is used in the comment, and to_user_id is the id of the target user of the user who posted the comment.

The reply_id in the reply table (Table 5) is the unique id of this table, and the comment_id is the foreign key associated with the comment table, indicating that the reply is to which paragraph of comments. user_id is the id of the user who posted the reply, and to_user_id is the id of the user who responded to @.

4.3. System Implementation. The physical structure of the system is described in terms of the system’s a physical network connection, server deployment, and user terminals. Figure 5 is a diagram of the system’s physical structure.

TABLE 1: User table.

| Field name | Field type | Illustrate | Constraint |
|--------------|------------|-------------------|------------------------------|
| User_id | Integer | Unique identifier | Primary key, cannot be empty |
| Name | Varchar | Username | Cannot be empty |
| User_gender | Integer | User's gender | Cannot be empty |
| Phone_number | Varchar | Phone number | Cannot be empty |
| Weixin_id | Varchar | User WeChat id | — |
| qq_id | Varchar | User QQ id | — |
| Password | Varchar | User password | Cannot be empty |

TABLE 2: Song list.

| Field name | Field type | Illustrate | Constraint |
|-------------|------------|---------------------------|------------------------------|
| Song_id | Integer | Unique identifier | Primary key, cannot be empty |
| Name | Varchar | Song name | Cannot be empty |
| Album_id | Integer | Belonging to the album id | Foreign key |
| Singer_id | Integer | Singer id | Foreign key, cannot be empty |
| Description | Varchar | Song description | Cannot be empty |
| Public_time | Datetime | Release time | Cannot be empty |
| Visitors | Integer | Views | Cannot be empty |
| Lyric | Varchar | Lyric text | Cannot be empty |

TABLE 3: Album list.

| Field name | Field type | Illustrate | Constraint |
|-------------|------------|-------------------|------------------------------|
| Album_id | Integer | Unique identifier | Primary key, cannot be empty |
| Name | Varchar | Album name | Cannot be empty |
| Singer_id | Integer | Singer id | Foreign key, cannot be empty |
| Public_time | Datetime | Release time | Cannot be empty |
| Visitors | Integer | Views | Cannot be empty |
| Description | Varchar | Album description | Cannot be empty |

TABLE 4: Comment form.

| Field name | Field type | Illustrate | Constraint |
|--------------|------------|---|------------------------------|
| Commented_id | Integer | Unique identifier | Primary key, cannot be empty |
| Song_id | Integer | The id of the song to which the comment belongs | Foreign key, cannot be empty |
| User_id | Integer | User id | Foreign key, cannot be empty |
| To_user_id | Integer | User id | Foreign key, can be empty |
| Public_time | Datetime | Issuing time | Cannot be empty |
| Floor | Integer | Floor number | Cannot be empty |
| Content | Text | Comments | Cannot be empty |

TABLE 5: Reply form.

| Field name | Field type | Illustrate | Constraint |
|-------------|------------|-------------------------|------------------------------|
| Reply_id | Integer | Unique identifier | Primary key, cannot be empty |
| Comment_id | Integer | Reply to the comment id | Foreign key, cannot be empty |
| User_id | Integer | User id | Foreign key, cannot be empty |
| To_user_id | Integer | User id | Foreign key, can be empty |
| Public_time | Datetime | Issuing time | Cannot be empty |
| Content | Text | Comments | Cannot be empty |

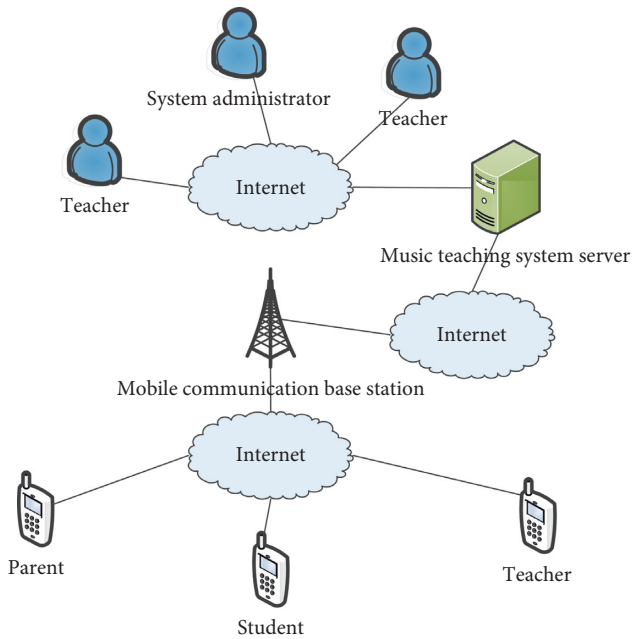


FIGURE 5: System physical structure diagram.

In the physical structure of the system in Figure 5, the music teaching server is the central server, providing mobile terminal application services for students, teachers, and system administrators. Students use mobile terminals to communicate on the server. Lecturers and system administrators can use the Internet to maintain the background data of the learning system.

The system development is completed under Windows 7, the integrated development environment is MyEclipse6.0, the development language is Java, the development kit is JDK1.6, and the web client uses HTML5+JavaScript to complete the development. There are two types of music homework: one is a self-study class with music homework, and the other is a resource for performance learning classes. Self-study lessons are not scored and are uploaded and maintained by system administrators and teachers. Performance variables are included in the evaluation scores, which can only be uploaded and maintained by teachers. Teachers upload music assignments according to the learning modules, and music assignments often take the form of file attachments. Learning information includes music assignments, editors, starting time, storage paths, etc. Students can find their own courses through the course classification or directly search for resources to get them. Users can design their own courses through management tools, or use learning tools to complete learning tasks. Music assignment information includes music assignment, publisher, release time, storage path, etc. Music assignments are posted by the teacher.

5. Conclusion

The online music teaching system is obviously different from the general teaching system. Music teaching has strong professional characteristics, such as the need to recognize

and pronounce notes. In order to maintain the best teaching effect, this paper proposes a music teaching system based on the Android platform. The system uses multimedia technology under the Android platform to provide a concise interface design for a teaching system with music professional characteristics. With the convenience and speed characteristics of today's mobile Internet, learners can make full use of their free time to complete learning and practice, thereby improving learning efficiency. At the same time, the research of this paper explores the frame structure design of a music teaching system based on distributed database and mobile terminal motion visualization. The music teaching system has been systematically tested and applied, allowing students to achieve complete learning of music through classroom learning, homework, and information notifications. It allows students to be in an online music learning environment, changing the traditional online teaching mode of learning only without practice, thereby maximizing the learning effect of learners.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

All the authors have no conflicts of interest.

References

- [1] R. Wang, "Design of online music teaching platform based on cognitive wireless sensor network and interactive design," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, pp. 1–11, 2021.
- [2] B. Allan and D. Lewis, "The impact of membership of a virtual learning community on individual learning careers and professional identity," *British Journal of Educational Technology*, vol. 37, no. 6, pp. 841–852, 2006.
- [3] H. Z. Lv and J. Luo, "Creative approaches in music teaching," *Thinking Skills and Creativity*, vol. 40, no. 1, 2021.
- [4] M. D. Schipor and D. S. Duca, "On-site vs online teaching: challenges and coping strategies," *Revista Romaneasca pentru Educatie Multidimensionala*, vol. 13, no. 1Sup1, pp. 12–26, 2021.
- [5] S. Wang and M. Mu, "Exploring online intelligent teaching method with machine learning and SVM algorithm," *Neural Computing & Applications*, vol. 36, no. 6, pp. 1–14, 2021.
- [6] E. Svirakova, R. Soukalová, P. Bednar, and L. Danko, "Culture managers education: system dynamics model of the co-working design centre," *Procedia - Social and Behavioral Sciences*, vol. 174, no. 106, pp. 1684–1694, 2015.
- [7] J. Fleck, "Blended learning and learning communities: opportunities and challenges," *The Journal of Management Development*, vol. 31, no. 4, pp. 398–411, 2012.
- [8] M. A. Akbar and I. Handriani, "Study and implementation information system of zakat using MVC architecture," *IOP Conference Series: Materials Science and Engineering*, vol. 453, no. 1, pp. 012032–12118, 2018.
- [9] S. C. Nistor and G. Czibula, "IntelliSwAS: optimizing deep neural network architectures using a particle swarm-based

- approach,” *Expert Systems with Applications*, vol. 187, p. 115945, 2022.
- [10] Y. Q. Yang and H. Chen, “Framework structure on enterprise office automation system,” *Applied Mechanics and Materials*, vol. 713-715, pp. 2246–2249, 2015.
- [11] G. Xun and L. Suxia, “Construction of evaluation system of sports talent training scheme based on data mining,” *International Journal of Reasoning-based Intelligent Systems*, vol. 10, no. 2, pp. 128–133, 2018.
- [12] S. Molano and A. Polo, “Social network analysis in a learning community,” *Procedia - Social and Behavioral Sciences*, vol. 185, pp. 339–345, 2015.
- [13] D. M. Kennedy, F. Yu, M. Yi, and Z. Xue, “Standards for online teaching: lessons from the education, health and IT sectors,” *Nurse Education Today*, vol. 25, no. 1, pp. 23–30, 2005.
- [14] X. Ning, “Design of online popular music teaching platform based on embedded computer network and virtual reality,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 35, no. 2, pp. 11–25, 2021.
- [15] A. Xw and C. B. Yao, “Music teaching platform based on FPGA and neural network,” *Microprocessors and Microsystems*, vol. 80, pp. 52–63, 2020.