Earth Sciences
2023; 12(4): 82-89
http://www.sciencepublishinggroup.com/j/earth
doi: 10.11648/j.earth.20231204.11
ISSN: 2328-5974 (Print); ISSN: 2328-5982 (Online)

Methodology Article

Design and Implementation of Database Management System for Mineral, Rock and Fossil Specimen Resources

Zhiyuan Wu[1, *], Mingyue He[2], Mei Yang[2]

[1]School of Information Engineering, China University of Geosciences (Beijing), Beijing, China

[2]National Mineral Rock and Fossil Specimen Resource Center, China University of Geosciences (Beijing), Beijing, China

Email address:

wuzy@cugb.edu.cn (Zhiyuan Wu), hemy@cugb.edu.cn (Mingyue He), yangmei@cugb.edu.cn (Mei Yang)

[*]Corresponding author

To cite this article:

Zhiyuan Wu, Mingyue He, Mei Yang. Design and Implementation of Database Management System for Mineral, Rock and Fossil Specimen Resources. *Earth Sciences*. Vol. 12, No. 4, 2023, pp. 82-89. doi: 10.11648/j.earth.20231204.11

Received: June 6, 2023; **Accepted**: June 26, 2023; **Published**: July 6, 2023

Abstract: According to the demand for standardized input, management and sharing of mineral, rock and fossil specimen data, based on the mineral, rock and fossil specimen description standards and specifications released by National Infrastructure of Mineral, Rock and Fossil for Science and Technology (NIMRF), a database management system for mineral, rock and fossil specimen resources was designed and developed. In this paper, the overall architecture of the database system is designed using the front-end and back-end separation model and three-layer structure, and then relevant description standards and specifications are integrated into the design of the mineral, rock and fossil specimen database. Finally, based on current popular open source frameworks such as Spring Boot, Vue, etc., the management system has been developed. It realizes management functions of adding, deleting, updating and querying for the mineral, rock and fossil specimen database and also includes function of system management. In this way, a uniform mineral, rock and fossil specimen resource database can be built, which lay a solid foundation for data sharing of mineral, rock and specimen resources on the NIMRF platform. The application of the database management system is helpful to promote material sharing by information sharing, and improve the utilization rate of existing specimen resources.

Keywords: Mineral Specimen, Rock Specimen, Fossil Specimen, Database, Management System, Spring Boot, Vue

1. Introduction

Mineral, rock fossil specimen resources are minerals, rocks, ores and fossil specimens collected, classified, identified, and studied by geologists for a long time, as well as all related data and research materials. They include not only physical specimens, but also data and research results related to mineral, rock, ore and fossils. Rocks, minerals, ores, and fossil specimens are all non-renewable resources and are characterized by scarcity. They were formed in different periods during the evolution of the earth, carrying the knowledge of geological scientific, and recording the evidence of scientific research [1]. Among them, many mineral resources are indispensable strategic materials for the development of national economy, and some precious fossil specimens are of great significance for studying the origin and evolution of life and determining the age of strata.

Geological museums, natural history museums, colleges, scientific research institutes and production units in China have collected and preserved lots of mineral, rock and fossil specimens, and have rich mineral, rock and fossil specimen resources. However, each resource unit has its own system. Due to the lack of uniform description standards and norms for mineral, rock and fossil specimen, the uneven level of digitalization, and non-systematic scientific management, etc., the effective sharing of specimen resources is greatly limited, resulting in low utilization of specimen resources and the inability to full play to the value of specimens. Therefore, the collection, sorting, preservation, utilization and sharing of mineral, rock and fossil specimen resources is an important, urgent, long-term and basic strategic task in China's geological science and technology undertaking [2].

For the above demands, relies on National Infrastructure of Mineral, Rock and Fossil for Science and Technology (NIMRF), this paper designs and develops a mineral, rock and fossil resource database management system according to the mineral, rock and fossil specimen description standards released by NIMRF platform [3]. It is hoped that by integrating relevant description standards and norms into the design of the specimen resource database, the system can realize the standardized input and normalized management of rock and mineral fossil resource data, and a uniform mineral, rock and fossil specimen resource database will be formed, which will serve as a solid foundation for the release and sharing of the mineral, rock and fossil specimen resources on NIMRF platform.

2. System Design

The development model with front-end and back-end separation [4] has become an industry standard usage method for internet project development. Front-end and back-end separation refers to a way to separate the front-end interface and back-end logic of a website or application. Compared with the traditional development model based on the MVC framework, the development model with front-end and back-end separation has advantages such as clear division of labor, improved development efficiency, and better Maintainability, easier expansion, better user experience, etc. This paper uses the front-end and back-end separation development model to design and develop the specimen resource database management system. The back-end of the system will be developed with lightweight J2EE technology stacks which include Spring Boot [5] and MyBatis [6], etc., and the front-end will developed with Vue. js [7] and Element UI [8]. The front-end is specially designed to utilize the JS library Axios [9] to intercept requests and responses, convert the data of request and response with json format automatically, etc. [10], and communicate with the back-end through the RESTful interface. Through the interactions, it can realize the decoupling of the front-end and back-end, improve development efficiency and system scalability [11].

2.1. System Architecture

Base on the development mode with front-end and back-end separation, the classic three-tier architecture is adopted to achieve high cohesion and low coupling for functional modules of the system. The overall system architecture is shown in figure 1.

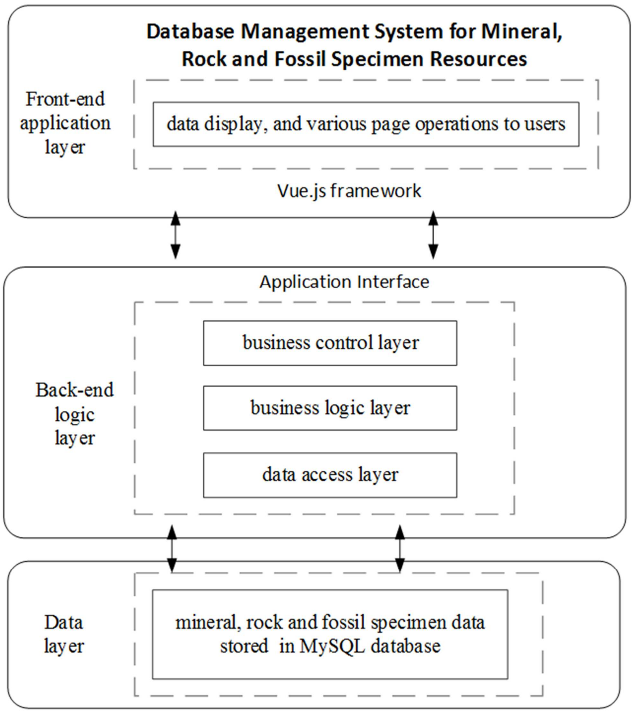

Figure 1. *System Architecture Diagram.*

As shown above, the system architecture is divided into data layer, back-end logic layer and front-end application layer from bottom to top.

Data layer: It provides data support for the whole system, including mineral, rock and fossil specimen data, resource classification code table, system dictionary table and other data, which are stored and managed uniformly by MySQL.

Back-end logic layer: Encapsulates the business processing logic of the back-end for mineral, rock and fossil specimen data. It reads data from the data layer downward, and provides interface services for the front-end application layer upward. The back-end logic layer should be developed based on the Spring Boot plus Mybatis framework. It also includes three layers: business control layer, business logic layer and data access layer.

Front-end application layer: directly to users. User manages the mineral, rock, ore, fossil specimen data through a web browser. When users operate on the web page, the front-end Axios library sends different requests to the back-end logic layer interface; back-end receives the requests and performs business logic processing, and then returns the response result to the front-end. The pages of front-end application layer are designed and built based on Element UI, which can help developers quickly build pages with powerful functions and unified style.

Table 1. *Structure of mineral, rock and fossil specimen database.*

Name	Description	Table
commonality database	commonality data table	BB_EPTDATA
	resource classification code table	BB_ZYGLBM
	core Metadata Table	BB_METADATA
personality database	rock personality data table	BB_P_YSSX
	mineral personality data table	BB_P_DXKW

Name	Description	Table
	ore personality data table	BB_P_JSKS
	fossil personality data table	BB_P_GSWHS
	Stone work personality data table	BB_P_SZP

The ER diagram of the relationship between all tables in the two databases is shown in figure 2.

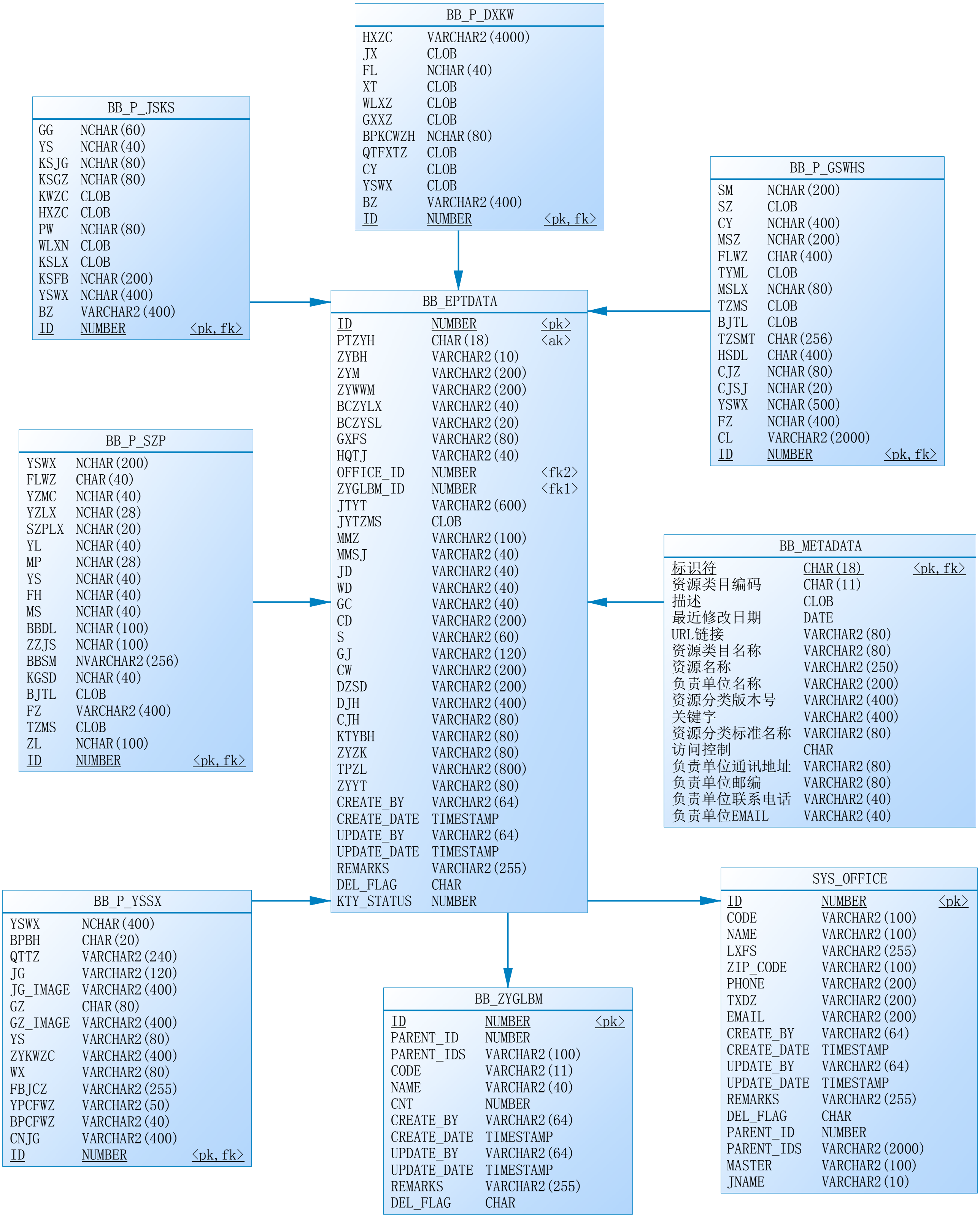

Figure 2. *ER Diagram of mineral, rock and fossil specimen database.*

2.2. Database Design

The design and development of the database system is an important part of the Java Web application system [12]. Whether the database design is reasonable or not, it directly affects the system performance, system maintenance and the function expansion. Database design should follow the principles which are mutual independence of data in the data table, low data redundancy, and database naming norms [13]. The basis for designing the resource database of mineral, rock and fossil specimens is the specification and standard of description for mineral, rock and fossil. According to the specification and standard, there are two databases can be designed which are commonality database, personality database.

2.2.1. Commonality Database

(i). Commonality Data Table

It can be seen from the figure 2 that the common data table BB_EPTDATA is the core table in the entire specimen resource database. It is designed according to the publicly released specifications for the common description of mineral, rock and fossil specimen resources. The table definition of BB_EPTDATA is shown in table 2.

Table 2. *Commonality data table.*

No	Field	Description
1	ID	Id number, primary key of the table
2	PTZYH	The resource number uniformly generated by the entire platform
3	ZYBH	National uniform numbering of the specimen
4	ZYM	The Chinese name of the specimen
5	ZYWWM	The foreign name of the specimen
6	BCZYLX	Type of the specimen preserved
7	BCZYSL	Number of preserved resources
8	GXFS	Sharing method
9	HQTJ	Access way
10	OFFICE_ID	The ID number of the associated unit table
11	ZYGLBM_ID	The ID number of the associated resource classification code table
12	JTYT	Specific use
13	JYTZMS	Description of the main characteristics of the specimen
14	MMZ	Provider of the specimen
15	MMSJ	The specific time when the specimen is provided
16	JD	The longitude of the specimen's place of origin
17	WD	The latitude of the specimen's place of origin
18	GC	The elevation of the specimen's place of origin
19	CD	The place where the specimen was collected
21	S	The province where the resource was collected
22	GJ	The country where the resource was collected
23	CW	The yield status of minerals, rocks, ore specimens, mainly refer to the production layer for fossils
24	DZSD	Geological age of resource formation
25	DJH	The number in the resource preserved unit
26	CJH	The number of the specimen when it was collected in the field
27	KTYBH	The number of the specimen in the specimen library
28	ZYZK	The state of the specimen
29	TPZL	Picture of this specimen
30	ZYYT	The main characteristic use of the specimen
31	CREATE_BY	The id of the user who created this data
32	CREATE_DATE	The time when the user created the data
33	UPDATE_BY	The id of the user who updated this data
34	UPDATE_DATE	The time when the user updated the data
35	REMARKS	Other instructions
36	DEL_FLAG	Mark whether this data is deleted
37	KTY_STATUS	Inventory status of the physical specimen

In table 2 above, the serial numbers 2-30 are the common description of the specimen data, and 31-37 are the auxiliary management information for the specimen data, which are used for the maintenance and management of the specimen data in the system.

(ii). Resource Classification Code Table

The resource classification code table BB_ZYGLBM is designed according to the classification and code table of mineral, rock and fossil specimen resources in the description standard. The table definition of BB_ZYGLBM is shown in table 3.

Table 3. *Resource classification code table.*

No	Field	Description
1	ID	Id number, primary key of the table
2	PARENT_ID	ID number of the parent

No	Field	Description
3	PARENT_IDS	All parent ID numbers, separated by commas
4	CODE	Code
5	NAME	Name of the code
6	CNT	Number of resources in coded range (including all ancestors)

As tree structure can well represent the hierarchical classification characteristics of resources, the resource classification code table is designed as tree structure. Table with tree structure has the advantages of small redundancy of data storage, strong intuitiveness, simple and efficient retrieval traversal. The PARENT_ID and PARENT_IDS fields of the resource classification code table link codes at all levels with their parent codes, and finally form a directory tree structure in database.

2.2.2. Personality Database

As can be seen from figure 2, there are five tables in the personality database: BB_P_YSSX, BB_P_DXKW, BB_P_JSKS, BB_P_GSWHS, and BB_P_SZP. They all have the ID column primary and foreign key association relationship with the common data table BB_EPTDATA. The commonality data and personality data of the specimen constitute the complete description information of the specimen.

The rock specimen personality data table BB_P_YSSX has a total of 14 fields, including slice number, other features, structure1, image of structure1, structure2, image of structure2, color, main mineral composition, physical properties, distribution and occurrence, storage location of sample, storage location of thin section, annual results of thin section, and literature.

The mineral specimen personality data table BB_P_DXKW has a total of 10 fields, including chemical composition, crystal structure, classification, morphology, physical properties, optical properties under microscope, location number of thin section inventory, other analytical features, origin, and literature.

The ore specimen personality data table BB_P_JSKS has a total of 11 fields, including specification, color, ore structure1, ore structure2, mineral composition, chemical composition, grade, physical properties, ore type, ore distribution, and literature.

The fossil specimen personality data table BB_P_GSWHS has a total of 16 fields, including genus name, genus, etymology, type species, systematic taxonomic position, heteronym, type, feature description, comparison and discussion, feature sketch, fossil measurement, collector, collection time, notes, materials, and literature.

The stonework product personality data table BB_P_SZP has a total of 18 fields, including site name, site type, stone product type, raw material, rough, color, weathering, abrasion, specimen measurement, production technology, specimen sketch, archaeological age, comparison and discussion, notes, character description, weight, bibliographic information, and taxonomic position.

2.3. Functional Design

According to the previous database design, the mineral, rock and fossil specimen resource management system mainly includes 6 functional modules, which can be shown in figure 3.

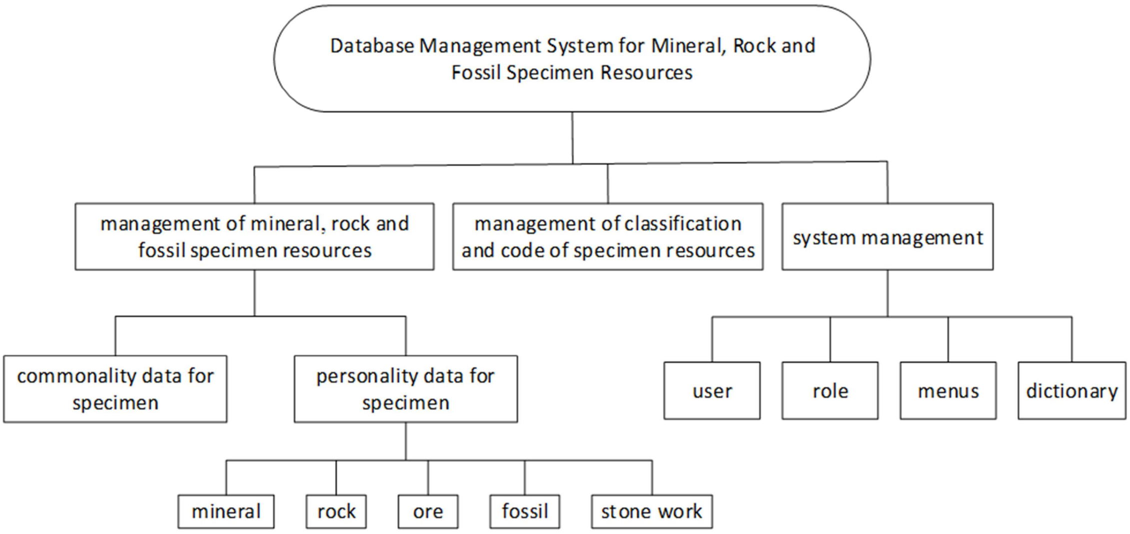

Figure 3. *Functional structure diagram of the system.*

2.3.1. Management of Mineral, Rock and Fossil Specimen Resources

This functional module includes the management of commonality data and personality data of mineral, rock and fossil specimen. A unified management interface is designed to provide the import, export, input, update, query and other operations for the commonality data of specimens. As stone work which belongs to the category of fossils has many different characteristics from other fossils, the personality data of mineral, rock, and fossil specimens are divided into five categories which are rock, ore, mineral, fossil and stone work. So, there are five function parts in this module for personality data. For each type of specimen, according to its personality characteristics, corresponding management operation interface is designed respectively to fulfill operations such as input, update, query, import, and export of the personality data. In addition, the system also includes the function of viewing and exporting the complete information of the specimen, which combines the commonality data and the personality data of the specimen together.

2.3.2. Management of Classification and Code of Specimen Resources

This module implements the system management of the resource classification code table in the common description specification, including code input, modification, deletion, query and other operations.

2.3.3. System Management

System information is maintained by system administrators. It includes four parts: user management, role management, menu management and dictionary management.

User management maintains basic user information (such as account number, password, and name, etc.) in the system, including adding users, modifying users, deleting users, querying users, and assigning user roles, etc.

Role management maintains the role information in the system, including role name, data range, column range, menu permissions, role user assignment, etc.

Menu management maintains the function menu information of the system, including creating function menus, modifying and deleting function menu. The function menu information includes menu names, menu link addresses, sorting, authority identification and other information.

Dictionary management maintains commonly used dictionary codes of the system, including operations such as adding, modifying, deleting, and querying. The dictionary code includes key value, label, type, and description, and is mainly used for reference by functional modules of mineral, rock, ore, fossil and stone work.

3. System Implementation

3.1. Technical Features

3.1.1. The Separation of Front-End and Back-End with Restful-Style

RESTful is an architectural pattern applied to internet software development. The lightweight interface designed with RESTful architecture can effectively solve the problem of data sharing between subsystems [14], improve system response speed, simplify program design, and improve overall system performance. In this paper, the backend of the system uses Spring Boot to develop interface with restful-style, and data format between the interactions of request and response through the interface is JSON. The front end sends API requests (GET, PUT, POST, DELETE, etc.) with RESTful style to the back end, and renders the page after obtaining the response JSON data.

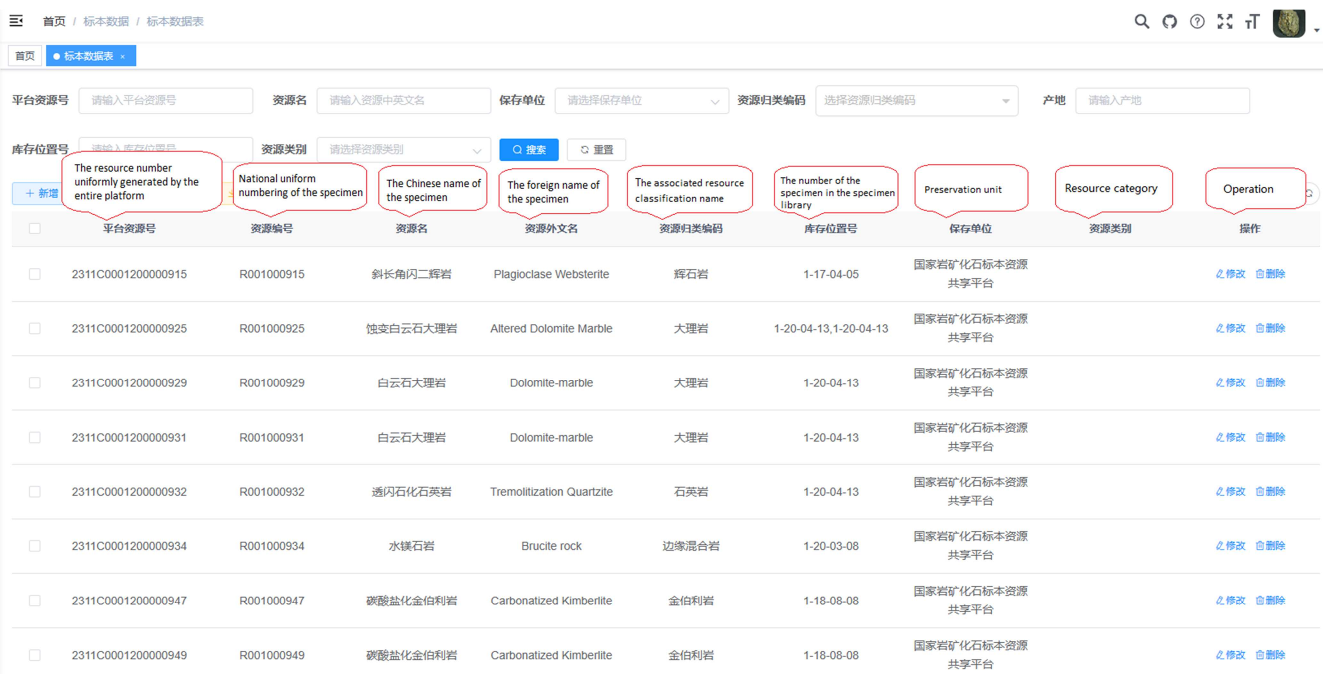

Figure 4. *Management web interface for commonality data of mineral, rock and fossil specimens.*

System administrator can manage resource classification and code which are displayed in the form of a tree structure, as shown in figure 5.

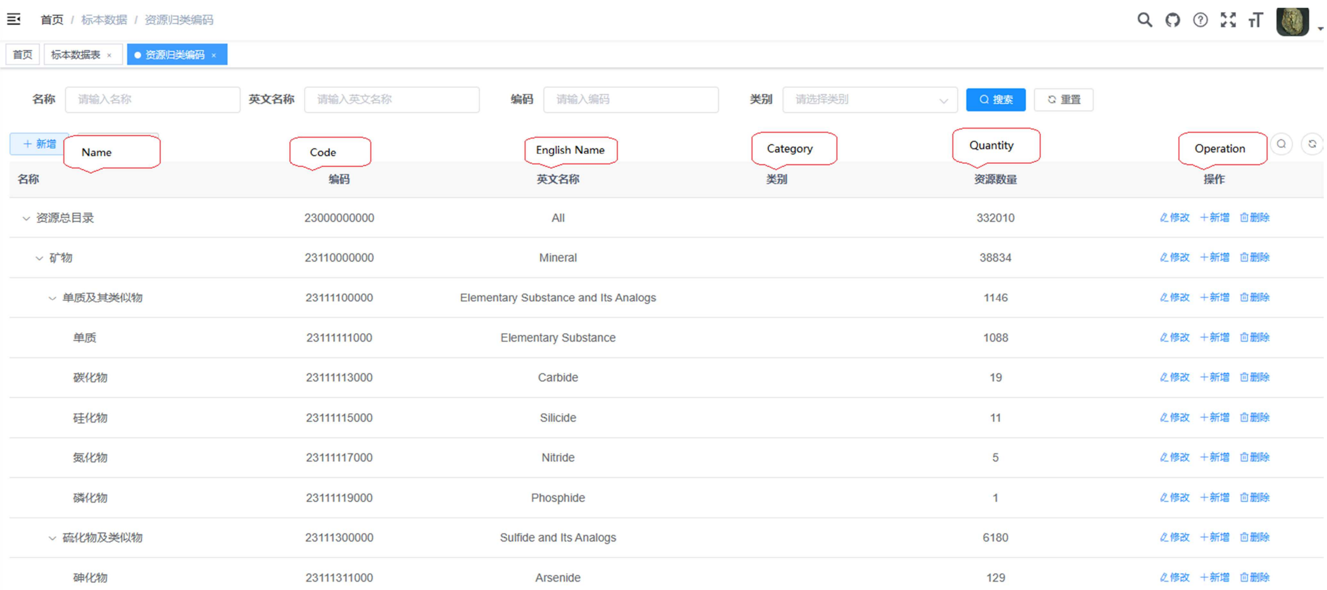

Figure 5. *Management web interface for classification and code of specimen resources.*

System administrators can add users, configure system menus and user roles, etc., as shown in figure 6.

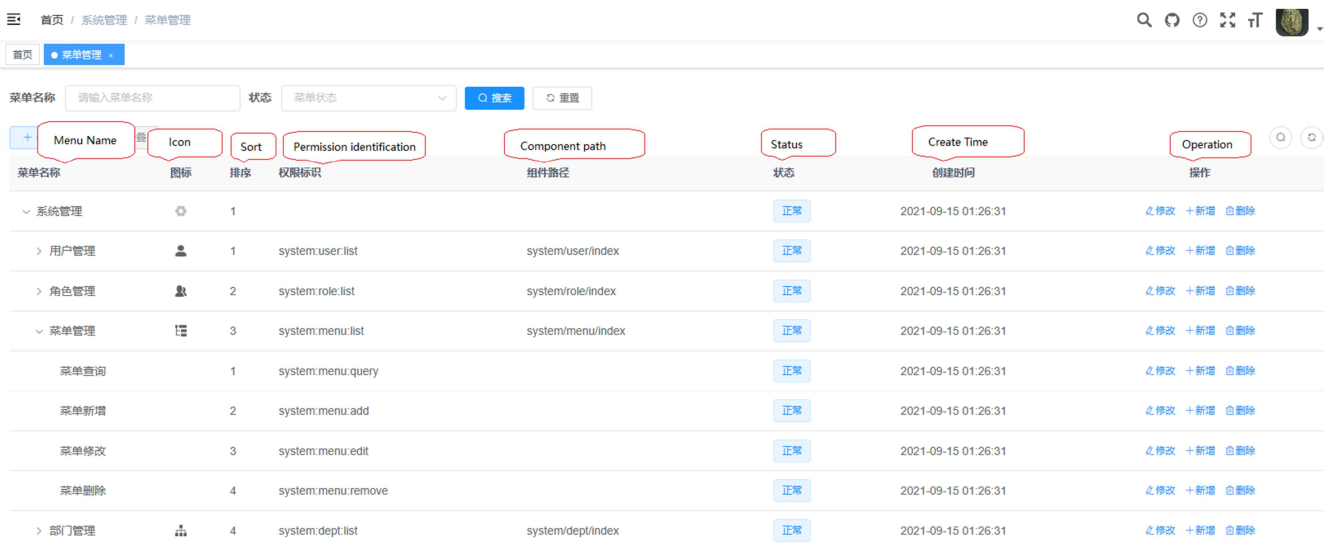

Figure 6. *Management web interface for system menus.*

3.1.2. Stateless Authentication and Authorization Using JWT

JWT, the full name is Json Web Token, is an open standard (RFC 7519) [15]. It is a JSON-style lightweight authorization and authentication specification, and is generally used to pass authenticated user identity information between identity providers and service providers in order to obtain resources from resource servers, as well as to add some additional declaration information necessary for other business logic [16]. The brief technical principle of JWT is as follows: a token is be generated when server verifies the login user's account and password, and will be returned to the client; the client will carry this token during the next visit, and the server is responsible for verifying the token every time, so stateless and distributed web application authorization can be realized. Traditional web applications usually use the Cookie-Session mode for user identity authentication, and the server's session object stores the user's authentication information, so the user's access to the service is considered safe. But it is only suitable for single application mode. The biggest problem with this model is that distributed architecture cannot be used, and it is not convenient for horizontal expansion. JWT stateless authentication and authorization can easily solve these problems. As mentioned earlier, the management system of mineral, rock and fossil specimen is designed using the front-end and back-end separation mode, and the back-end provides rest-style interfaces. One of the most important specifications of the rest-style is the stateless nature of the service, so JWT token is adopted to provide the authentication and authorization for system users.

3.2. System Function Display

According to the previous system design, languages such as Java, Html, JS, CSS, and related development frameworks were used, and the database management system for mineral, rock and fossil specimen resources was developed and constructed. After logging in to the system, data administrator can perform management by various operations such as input and modification of mineral, rock, and fossil specimen data according to the permissions granted by the system, as shown in figure 4.

4. Conclusion

Based on the description standards and specifications of mineral, rock and fossil specimens released by NIMRF, this paper develops and builds a mineral, rock and fossil resource database management system using the current popular open source framework. In the paper, mineral, rock and fossil specimen resource database is first designed based on the description standards and specifications for mineral, rock and fossil specimens. Then the open source frameworks such as Spring Boot, Vue, etc. are used, combined with the front-end and back-end separation mode. The functional management modules of commonality data and Personality data of rock, ore, fossil, stone work specimens are developed and realized respectively. The construction and application of the system will help to the standardization of mineral, rock and fossil specimen resource data, and build a solid foundation for data sharing of mineral, rock and specimen resources on the NIMRF platform. On the basis of the existing standardized database, further data mining and analysis of the mineral, rock and fossil specimen resources can be carried out in order to give full play to the value of the data. The system is applied in the field of rock and mineral fossil specimens, and can also provide reference and demonstration for the data management of other specimen resources.

References

[1] CHEN Yuan, YU Bisheng, YUAN Wei. (2003). Development of Geologic Specimen Database Based on Web [J]. Computer Engineering, 5: 181-183.

[2] About NIMRF [EB/OL]. http://www.nimrf.net.cn/pub/jianjie.jsp. 2023, 06.

[3] Description Standard [EB/OL]. http://www.nimrf.net.cn/pub/msbz.jsp. 2023, 06.

[4] Fang Sheng, Qin Xiaoan, Wang Rui. (2021) Journal of Yellow River Conservancy Technical Institute, 33 (4): 50-54.

[5] Spring Boot [EB/OL]. https://spring.io/projects/spring-boot. 2023, 06.

[6] MyBatis SQL mapper framework for Java [EB/OL]. https://blog.mybatis.org/. 2023, 06.

[7] Vue.js - The Progressive JavaScript Framework [EB/OL]. https://vuejs.org/. 2023, 06.

[8] Element - A Desktop UI Toolkit for Web [EB/OL]. https://element.eleme.io/. 2023, 06.

[9] Axios - Promise based HTTP client for the browser and node.js [EB/OL]. https://github.com/axios/axios. 2023, 06.

[10] LI Hang, DONG Anming, YU Jiguo, HAN Yubing. (2022) Design of intelligent agricultural IoT system based on front-rear end separation architecture [J]. Modern Electronics Technique, 45 (14): 63-68.

[11] WANG Jian, LUO Zheng, ZHANG Xi, ZHANG Mengqi, ZHANG Ke. (2020) Design and Implementation of WEB Project Front and Back Separation [J]. Software Engineering, 23 (4): 22-24.

[12] ZHONG Defu, ZHANG Liangguo, AI Hong, HUANG Xiaohua, LYU Junlin. (2019) Framework reconstruction of fishery resource survey data acquisition system based on NodeJS [J]. Fishery Modernization, 46 (6): 104-109.

[13] Wei Jia'nan, Wu Yong, Lin Huajian, Gong Yilong, Song Yun, Fu Junhao. (2020) DESIGN AND IMPLEMENTATION OF "ONE MAP" MANAGEMENT INFORMATION SYSTEM FOR NATURAL RESOURCES BASED ON WEBGIS [J]. Computer Applications and Software, 37 (9): 20-26.

[14] SHAO Jian-wei, LIANG Zhong-min, WANG Jun, HU Yi-ming, LI Bin-quan. (2020) Design and Development of Medium and Long-term Hydrological Forecasting System Based on SpringBoot Framework [J]. Water Resources and Power, 38 (4): 6-9, 5.

[15] XIANG Wu-ming, BAO Liang, YU Shao-hua. (2018) Design of Role Authentication Scheme for RESTful API Based on JWT [J]. Modern Computer, 34: 82-85.

[16] CHEN Jia. (2021) A Microservices Authentication and Authorization Scheme for Power System Based on JWT Token-based Authentication [J]. Electric Engineering, 16: 151-154.