ISSN: 2710-0588

Design and Implementation of Ship Shore Power Information Management System based on Spring Boot and Vue

Yujuan Zou[1, *], Peiyi Tang[2], Taizhi Lv[2]

[1]School of Information Technology, Jiangsu Maritime Institute, Nanjing, China

[2]R&D Department, Nanjing Huihai Transportation Technology Company Limited, Nanjing, China

*54552046@qq.com

Abstract

Ship shore power is an effective means to reduce air pollution in ports, which has been rapidly popularized in China ports. With the rapid increase of the number of shore power facilities, the monitoring and management details of shore power facilities cannot meet the requirements, and the focus of the promotion of shore power is shifted to management. In order to better manage the shore power facilities and better serve the use of ship shore power, a ship shore power information management system is designed and implemented. This system is based on the Spring Boot and Vue framework, and is implemented through technologies such as front-end and back-end separation, mixed APP, and multi-level storage architecture. The application of the system effectively supports the high-quality development of ship shore power, and the data acquisition through the supervision module can scientifically guide the construction of ship shore power facilities.

Keywords

Ship Shore Power; Front-end and Back-end Separation Technology; Spring Boot; Multilevel Storage Architecture; Vue; Uni-app.

1. Introduction

Shore power is the most effective means to reduce atmospheric pollution emissions during ship berthing, and it is an important means to promote China's strength in transportation [1-2]. In recent years, with the strong promotion of local governments and shipping enterprises, the number of berths with power supply capacity has made a qualitative leap. With the increasing number of shore power facilities, how to manage and use shore power facilities well has become the focus of shore power promotion. In order to support ports and shipping enterprises to strengthen the all-round management of shore power equipment and facilitate the maritime supervision department to master the use of shore power facilities in real time, this paper designs and implements a ship shore power information management system. On the one hand, the establishment of the system is conducive to the unified management and scheduling of the use of port ship shore power facilities, improving the use efficiency of ship shore power, and improving the shore power management level of port and shipping enterprises. On the other hand, the establishment of the system can also provide the maritime supervision department with a real-time grasp of the use of shore power facilities, and provide a basis for the layout and policy formulation of shore power facilities through big data analysis.

The ship shore power system adopts the front-end and back-end separation technology. The back-end is realized through Spring Boot, and JSON format data is provided to the front-end through Restful interface. The front-end realizes data access, binding and display through Vue

ISSN: 2710-0588

framework. The system is more stable and scalable through the front-end and back-end separation technology.

2. System Design

2.1. Functional Design

The system users are mainly divided into crew users, ship shore power enterprises and maritime regulatory agencies. The crew users can realize the whole chain management of ship power use, reservation, fault report and power query through the mobile terminal. Ship shore power enterprises manage the use of shore power facilities through the enterprise workbench. The main functions include adding non intelligent shore power equipment, viewing the information of all shore power equipment under the operation site of the enterprise, downloading the QR code of shore power equipment for crew scanning and charging, reviewing crew's shore power reservation orders and ship's shore power query and statistics. Maritime supervision users can view the use of shore power within their jurisdiction through the supervision workbench, grasp the trend and characteristics of the use of shore power facilities through data analysis and visualization functions, more effectively ensure the scientific use of ship shore power, and optimize the layout of ship shore power piles. The system functional module diagram is shown in Figure 1.

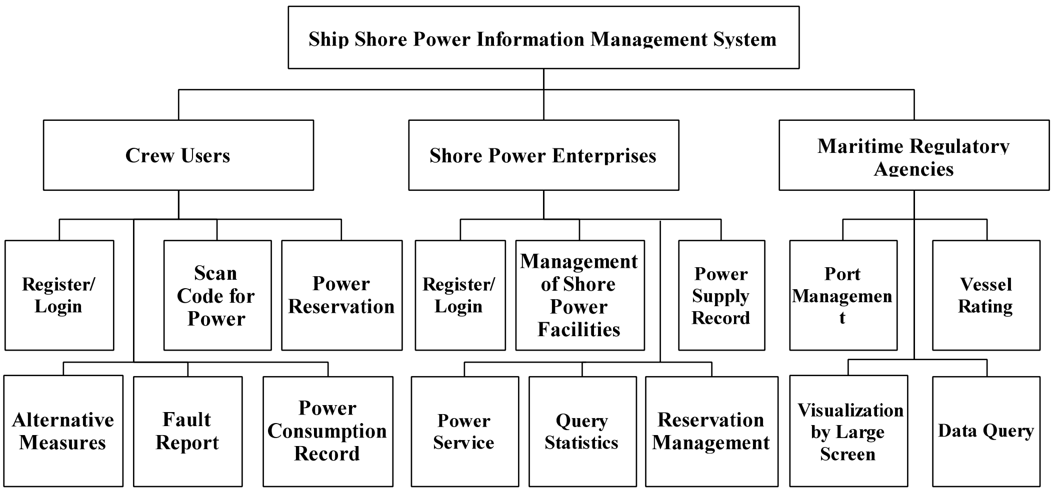

Figure 1. The functional module diagrams

2.2. Architecture Design

The core idea of the front-end and back-end separation technology is that the front-end pages interact based on JSON format data through AJAX calling the back-end connection [3-4]. As shown in Figure 2, the system is implemented by front-end and back-end separation technology, which is divided into front-end, back-end and data end to effectively decouple. Each end is deployed on different servers, which improves the reliability of the system. The front-end is deployed on the nginx Web server and is responsible for parsing static resources. The back-end is deployed on the application server to realize dynamic resource analysis and data query, and the reliability of back-end services is improved through elastic computing architecture. The data side adopts a multi-level data storage architecture, including Redis cache, MySQL database and Doris data warehouse, to meet the data access requirements of the system.

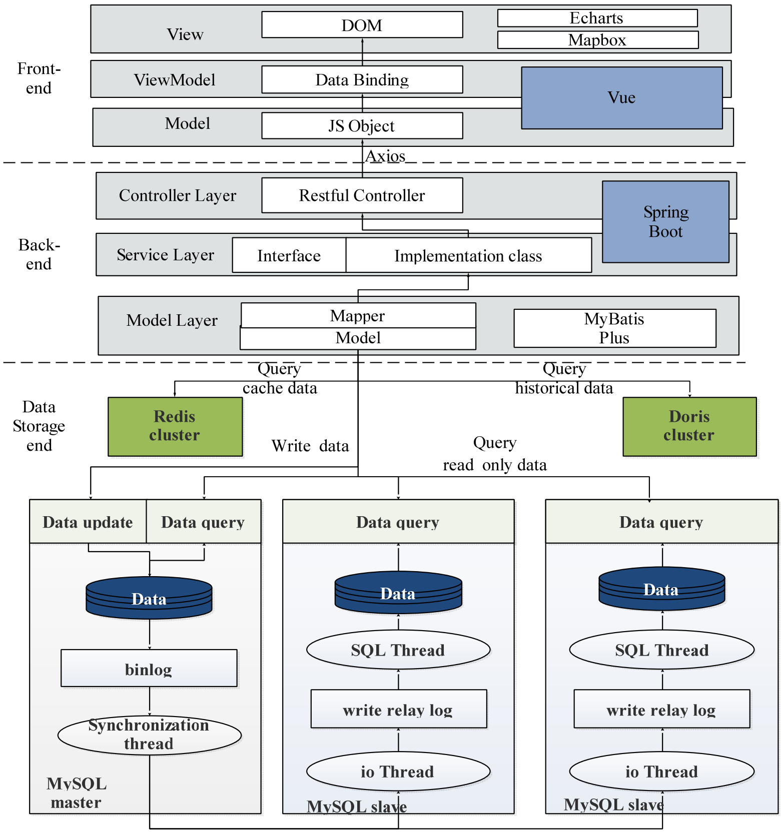

Figure 2. The system architecture

3. System Implementation

3.1. Back-end Implementation

{"features":[{"geometry":{"coordinates":["120.75614","32.031915"],"type":"Point"},"id":"9","type":"Feature","properties":
{"icon":"occupy"}},{"geometry":{"coordinates":
"120.552484","32.056069"],"type":"Point"},"id":"25","type":"Feature","properties":{"icon":"occupy"}},{"geometry":
{"coordinates":["120.756648","32.031637"],"type":"Point"},"id":"26","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":["120.477616","32.118687"],"type":"Point"},"id":"39","type":"Feature","properties":
{"icon":"occupy"}},{"geometry":{"coordinates":
"120.552484","32.056069"],"type":"Point"},"id":"50","type":"Feature","properties":{"icon":"occupy"}},{"geometry":
{"coordinates":["121.018144048965","31.7517828801765"],"type":"Point"},"id":"51","type":"Feature","properties":
{"icon":"occupy"}},{"geometry":{"coordinates":
"120.73407486976033","32.03953439073135"],"type":"Point"},"id":"58","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"121.01761413487459","31.796849623154014"],"type":"Point"},"id":"68","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"121.06406633584837","31.797134758974863"],"type":"Point"},"id":"69","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"120.75610024615071","32.032089504575325"],"type":"Point"},"id":"72","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"121.7277374976531","31.714993396533032"],"type":"Point"},"id":"78","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"120.96277770467415","31.77198836754701"],"type":"Point"},"id":"85","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":["120.6828","32.0078"],"type":"Point"},"id":"115","type":"Feature","properties":
{"icon":"occupy"}},{"geometry":{"coordinates":
"120.92122277772009","31.89498652329094"],"type":"Point"},"id":"120","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":
"120.53713843362746","32.08777793192027"],"type":"Point"},"id":"126","type":"Feature","properties":{"icon":"occupy"}},
{"geometry":{"coordinates":["120.477616","32.118687"],"type":"Point"},"id":"143","type":"Feature","properties":

Figure 3. The equipmentsLoc interface

The back-end is composed of control layer, services layer and model layer. The control layer is implemented by the back-end Spring Controller. The controller receives the front-end request, calls the business logic layer, and returns the processing results to the front-end. The services layer realizes shore power equipment analysis, shore power data statistics and basic configuration, and the model layer is implemented by MyBatis plus to complete the interaction

ISSN: 2710-0588

with the database. In order to improve data access efficiency, a multi-level storage architecture is built, including Redis cache, database and data warehouse. For high-frequency access data, including user information, ship status and other information, Redis improves system efficiency. For shore power data synchronization to two MySQL slave databases, the shore power data can be loaded within 3 seconds without affecting the operation of the user's shore power equipment. The ship track information is stored in Doris data warehouse, and massive data can also be queried in seconds.

The query interface equipmentsLoc for querying the coordinates of ship shore power facilities is taken as an example. This interface first defines the equipmentsLoc method in the EquipmentController controller, and defines the interface access address as equipmentsLoc through annotations. This method queries power facilities from the table quipmentLoc obtains the coordinates of the shore power facilities through Mybatis plus. It then converts the query results into JSON format to provide them to the front-end. The data format is shown in Figure 3.

3.2. Front-end Implementation

The front-end is based on MVVM mode and implemented by the visualization layer. Vue is a progressive JavaScript framework based on MVVM mode, which can quickly build front-end applications [5]. In order to display the use of shore power more intuitively through the map, the system uses Mapbox to display the map. Mapbox is a geographic information data visualization tool, which realizes the map visualization function by overlaying the shore power layout layer on the Mapbox map layer [6]. The front-end sends the request based on the Axios asynchronous request, the back-end returns JSON data, the data is bound to the model, and synchronized to the Mapbox visualization chart through the ViewModel.

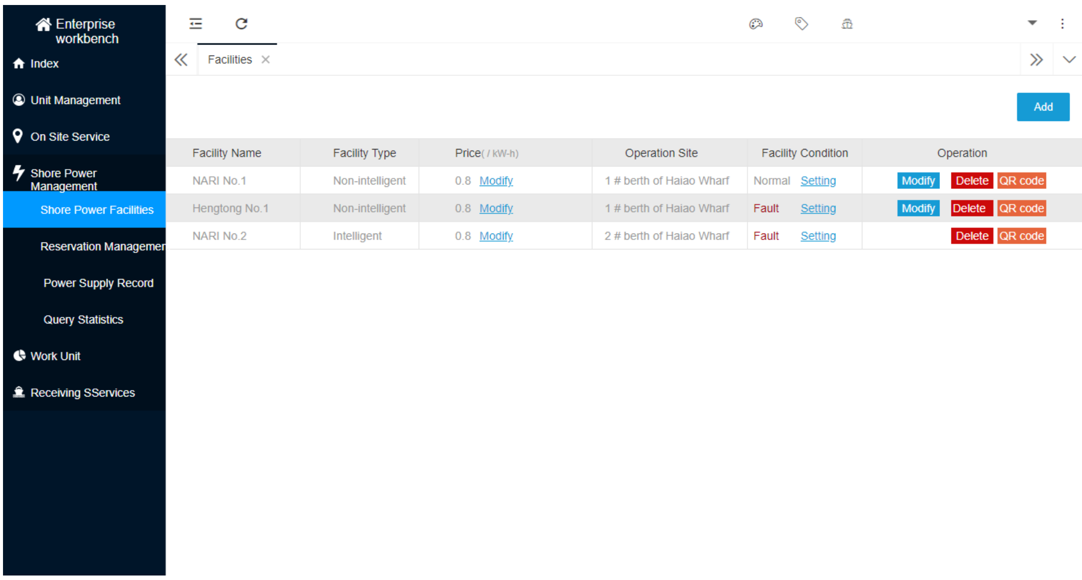

Figure 4. The power facilities management page

3.3. Mobile-end Implementation

The mobile end is developed by uni-app, which can be released to Android, WeChat applet, IOS and other platforms through a set of codes, reducing the development cost of the mobile terminal. The uni-app framework is based on Vue development, with the advantages of a large number of cross end, excellent performance experience, rich ecological openness and low development costs, has been widely used in mobile terminal development. Uni-app is basically implemented in the same way as the front-end, sharing the same back-end interface. The mobile end requests back-end data through Axios, and then renders it to the mobile terminal interface through uni-app. As shown in Figure 4, it queries the power consumption of a user through the getPowerUsage interface, and then render two power consumption records to the page.

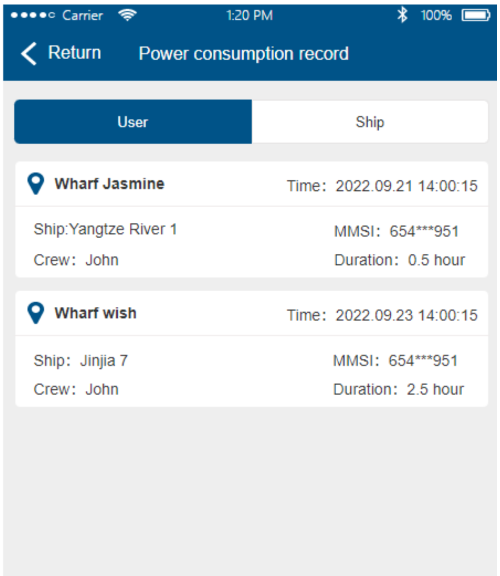

Figure 5. The power consumption record of the mobile end

4. Conclusion

With the deepening of global economic integration, national trade exchanges are increasingly frequent, and the throughput of ports in various countries has reached new highs. Port economy brings great economic benefits to port cities, but it also brings heavy burdens. In order to promote energy conservation and emission reduction and improve comprehensive competitiveness, many ports have built ship shore power facilities to enable ships to obtain power during berthing in ports, thus stopping the ship's own generators and reducing emissions. The construction of ship shore power facilities has achieved remarkable results. In order to improve the utilization rate of ship shore power and manage the shore power facilities more effectively, it is necessary to establish a ship shore power information management system. Based on this, this paper designs and implements the ship shore power information management system. The system adopts the mainstream front-end and back-end separation technology, the back-end uses Spring Boot to provide data interfaces, the front-end uses Vue technology to achieve data rendering and collection, and the mobile-end uses uni-app to achieve development across Android, IOS, WeChat applet and other platforms. The data side adopts a multi-layer architecture to meet different data requests.

Acknowledgments

This work was financially supported by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (21KJB580007), the Excellent Scientific and Technological innovation Team of the Jiangsu Higher Education Institutions of China (Maritime big data team), the Young Academic Leaders for QingLan Project of the Jiangsu Higher Education Institutions of China.

ISSN: 2710-0588

References

[1] Ye, Guanqiong, et al. "Are shore power and emission control area policies always effective together for pollutant emission reduction?–An analysis of their joint impacts at the post-pandemic era." Ocean & Coastal Management 224 (2022): 106182.

[2] Yin, Wenwei, et al. "Shore power management for green shipping under international river transportation." Maritime Policy & Management 49.5 (2022): 737-754.

[3] Kang, Xiaolin, et al. "Design of student information management system based on the separation of front and back ends." 2nd International Conference on Internet of Things and Smart City (IoTSC 2022). Vol. 12249. SPIE, 2022: 103-109.

[4] Zhang, Juan, Peiyi Tang, and Taizhi Lv. "A real-time calculation platform of ship air pollutant emission based on Flink." 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE, 2022: 785-788.

[5] Bielak, Konrad, Bartłomiej Borek, and Małgorzata Plechawska-Wójcik. "Web application performance analysis using Angular, React and Vue. js frameworks." Journal of Computer Sciences Institute 23 (2022): 77-83.

[6] Tang, Peiyi, and Taizhi Lv. "Design and Implementation of Ship Water Pollution Supervision and Service Information System." Academic Journal of Computing & Information Science 5.11: 85-89.

[7] Wang, Yanfeng, et al. "Military chain: construction of domain knowledge graph of kill chain based on natural language model." Mobile Information Systems 2022 (2022): 1-11.