

RESEARCH ON KEY TECHNOLOGY OF BACKGROUND MANAGEMENT VISUAL PAGE CONSTRUCTION SYSTEM BASED ON COMPONENTIZATION

Zhenlin Huang^{1*}, Xing Wen¹

¹China Southern Power Grid Corporation Limited EHV Transmission Company

Guangzhou, Guangdong 510700, China

E-mail: huangzhenlin@ehv.csg.cn

Abstract - In the traditional front-end web page construction process, there are often many problems restricting efficiency. In order to improve the development efficiency, automatic development has become an important direction of front-end page development, and the automatic construction of the background management system page is conducive to solving the drawbacks of low efficiency of such webpage development, which is of great significance to the development of the Internet. Therefore, this article based on the modular development way, build the system background management visualization page for the design and implementation of research, and the implementation of system, the inspection result discovery system in different environment, the rendering time all can reach the ideal state, although there is some delay, but in general can satisfy the actual needs. At the same time, under the action of JEST testing framework, the routing address and navigation data can be printed in the terminal, which confirms the efficiency of the system module function. In addition, the system interface was previewed with the help of different devices, and the visual sense was found to be good. Thus, the feasibility of the system is confirmed and can be implemented in real life.

Keywords: Componentized development, Background management, Visual page, System, Design.

1. Introduction

With the advent of the Internet era, many enterprises have an increasing demand for Web. However, in the past, the development of Web pages needed to start from scratch, which was not only time-consuming and profitable, but also required the support of HTML, JavaScript and other knowledge [1-3]. In addition, in the process of web page construction, the user interface prototype design needs to be processed by artists, followed by programming to make it a static web page, and finally, the use of well-written logic code to generate dynamic pages. During this period, users' needs should also be taken into account for real-time modification. Therefore, there are many problems affecting efficiency during web development. At this point, the componentized development mode emerges at the historic moment and becomes an important direction of front-end development [4-9]. Based on this, on the basis of componentization, this paper will complete the design and implementation of the background management visual page construction system, and carry out the evidence, in

order to solve the drawbacks of front-end page development.

The system can screen Web page code, can simplify the front-end development process, reduce the developer development difficulty, to build a management page to meet the functional requirements. During the research, due to the length of the project, we omitted the detailed design of the project packaging and downloading function module, back-end function module and other modules in the design and implementation stage, and chose more representative function modules for design and implementation, which may be supplemented in the follow-up research process. Therefore, the writing of the project has certain limitations. Moreover, limited hardware configuration in practice hindered the progress of the project. However, with unremitting efforts, the project was finally completed, which is also the limitation of research resources. To get to the point, although there are some deficiencies in the subject research, the overall research is of practical significance, which can greatly improve the development efficiency and promote the development of enterprises.

2. Core Technology Analysis

In React, componentization is an important part of the concept [10]. Today, Web components are the most fundamental element of a Web application'S UI. In other words, React builds the user interface based on components' architectural approach. In addition, the interface can be decomposed into interacting components and, if necessary, components can continue to be decomposed into simple component combinations. When creating a new component, you are faced with different choices, such as a class or a stateless function component. Class-based components are a common way to create components in React if an application needs to use data state. You can either embed a subcomponent in a class component or use function methods to create components in React. In general, React componentization has four advantages. First, it is reusable. After defining components, you can reuse them in corresponding functions to reduce code duplication and enhance the logic of the code. Second, easy testing. If a component is faulty, it can be repaired independently without affecting other component tests. Third, simple maintenance. In terms of maintenance, a component can be maintained separately, reducing the difficulty of Web application maintenance. Fourth, have the ability to introduce third party components, and be able to use robust components in their own applications.

3. System Function Design

3.1 Page Editing Function Module Design

This module mainly includes the operation of the system page layout. The function of page editing module can first visually edit the navigation menu of the system, automatically map routes through the Settings of configuration data, and generate the navigation menu of the system. In addition, the navigation menu has interactive function. After the user clicks the navigation menu component, the user can jump to the page according to the bound route. It is worth noting that each navigation menu and its corresponding page of the system need to have access to the setting function. The second is the function of dragging and dropping components on the page, zooming, dynamically adding new components, and deleting components. The components loaded to the current page, its position x, Y value on the page, as well as width, height and other properties for real-time monitoring and rendering, so that users can freely arrange and combine components on the interface. Finally, the page adopts the responsive layout of streaming grid. Responsive layout design can provide users with a

better browsing experience, making the page suitable for a variety of terminals. The system has adaptive cross-device response function, which can make the layout of the page adaptively change with the user's environment.

3.2 Functional Module Design of Component Library

Considering the reusability of component design, JSX language is used as a language extension of Java Script. There are certain differences between JSX language and HTML language. The former tag can correspond to multiple components, and then there is no component tag. So, you can render multiple components on the interface by making multiple calls to the component using the JSX tag. However, the development of components needs to meet the following requirements: First, high cohesion, that is, functional logic code Java Script, pictures, static resources HTML unified encapsulation, so that it becomes a component. Second, reusability. When the component coupling is weak, the clearer the component attribute is, the stronger the component application will be and the higher its reusability will be. Third, interactivity, that is, interoperable logical functions (data deletion, paging display, etc.) are included in the table class components, so that components can interact with data, and display data in a dynamic way. Based on the appeal request, the development of the component library is carried out on the basis of Ant Design and Echarts component style library. Specifically, first of all, according to the different functions, the Java Script code is divided, and then the style and logic are encapsulated into independent front-end components that can interact, to achieve the construction of the component library. Secondly, the common methods such as data item deletion, modification and addition are encapsulated to improve the logicity of the code and the callability between functions. Through this operation, the code repeatability can also be reduced. Finally, the specific types of system components are designed. For the background management system of The Internet application system, the functions are similar and relatively fixed. Therefore, Echarts ICONS can be used for the data of the chart components to achieve data display and analysis. With Ant Design to achieve table components (data information increase, modify, etc.) development. In summary, the functional design of component library can be summarized as Figure 1, as shown below:

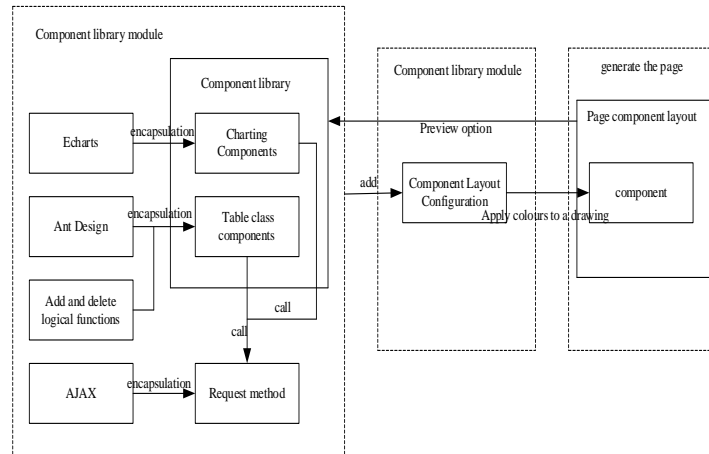


Figure 1: component library function design

It can be seen that the user can browse and select the components of the component library, and the system will render and draw the components selected by the user on the page for the user to use. It is worth noting that the logic code of the interactive function will be encapsulated in the component together, and the function code related to the AJAX request back-end interface to obtain data will be encapsulated into a public request method, so that the component has the function of interacting with the user, and can Visually request the interface to display dynamic data. It can build dynamic pages that can interact with the background data interface and meet various functions required by users, making up for the deficiencies of most of the current page building tools.

3.3 Component Editing Function Module Design

Component style editing relies primarily on component parameterization. If the parameterization of a component is regarded as a function, it can be understood that when different parameters are input, the function can return different operation results according to the different input parameters. Similarly, components can set various properties of components by passing parameters down from the upper component. Component editing function module mainly realizes dynamic visual editing of component attribute parameters, and its design diagram is shown in Figure 2.

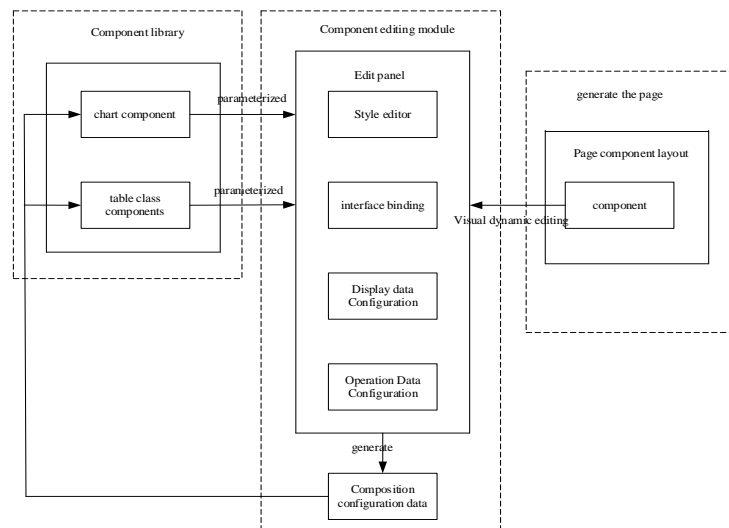


Figure 2: Design of component editing function module

As you can see, visual dynamic editing includes component title, component label and other style attribute parameter configuration. It provides various functions such as binding back-end interfaces of components, configuring interface parameters, configuring display data of components,

and configuring operation item data. Users can configure parameters visually by using component editing module. In this mode, components can be dynamically modified and edited without repeated debugging of code details, and components that meet requirements can be constructed.

3.4 Configuration Data Operation Module Design

To improve the robustness of the system and save and read system configuration data, the Local Storage browser is also used. In the design of configuration data operation module, the system data stored by the browser and database can be used for the memory layout function of the system, and the system configuration data can be stored locally in the browser in real time. In addition, you can manually upload configuration data to the database to ensure persistent storage of system data. For local browser storage and read, in order to avoid the browser abnormal close the wrong operation, lead to user configuration data missing problem, and for users to design operation page configuration data and real-time automatic save function, component configuration data that the user action of page components, location layout, size, and so on will be real-time monitoring system configuration data change events, Automatically generates configuration data and automatically stores the configuration data in the browser. When the user runs a large amount of Java Script code in the browser, the configuration data can be stored locally in the browser. Generally, the maximum Storage space of the Local Storage of a browser is 5M, which meets the requirements for storing system configuration data. Table 1 shows the Storage space of mainstream browsers.

Table 1. Storage space disclosure of major browsers

browser	chrome	firefox	IE
version	87.0	86.0	11.0
Local Storage	5m	5m	3120kb

The specific steps of configuration data operation module design are as follows: First, call the browser to store Local Storage related interface; Second, the system configuration data is saved to the Local Storage of the browser according to the JSON data format. At this point, the configuration data operation module is designed. When the user opens the browser, the browser will load the configuration data in the Local Storage and render and draw the page content according to the result saved last time. In terms of database persistent Storage and reading, the system studied in this paper sets the Storage and reading of official database data files to be uploaded and read only through users' manual operation in order to distinguish the data types stored in the Local Storage of the browser. In this way, through the different design of file storage, the user's system editing data can be saved flexibly locally, as well as the persistence of the database. For example, when the user closes the browser window, the popup

window can be designed to remind the user whether to save the configuration data. This also prevents the user from closing the browser window without saving their work. And the design page can read the data stored in the database. Users can choose whether to load the data files saved by the database. When the user chooses to load the database data, the configuration data operation module requests and reads the latest configuration data saved in the database through the interface. Otherwise, the configuration data operation module reads the data of the latest version saved from the Local Storage of the browser and renders the data on the interface.

4. System Function Realization

4.1 Page Editing Function Module Implementation

In this study, Web Storm is chosen as the integrated development environment for the front-end development of the system, IDEA is chosen as the integrated environment for the back-end development of the system, and the code management and version control are realized by using Tortoise SVN. The operating system and browser environment in the development process mainly use Windows and Chrome browser. According to the design above, the page editing function module mainly realizes the functions including the visual editing function of the navigation menu of the system and the setting of the access right of the page. At the same time, components can be dragged, placed, zoomed, added, deleted and other functions. Users can arrange and combine components freely on the interface. The page layout of flow grid is realized, and the page has adaptive cross-device response function. In addition, in order to complete the overall framework Layout of the system, also used the Pro Layout component, compared with the conventional component, Pro Layout component is very heavy, in which integrated navigation menu, Layout, page header, drawer and other functional components. The interface you see using the Pro Layout component in Ant Design Pro is shown in Figure 3.

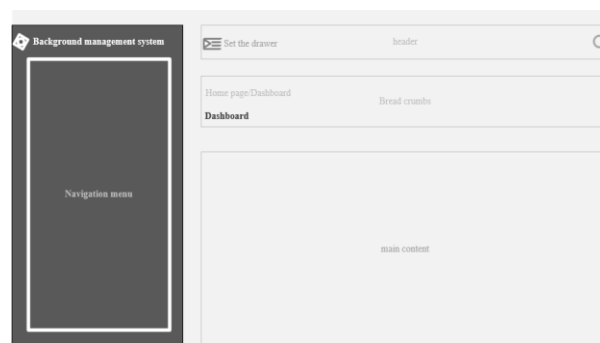


Figure 3: Overall layout of the page

For a page, its key skeleton is organized by the navigation menu. In order to facilitate management, the navigation menu first config.ts for the configuration and management of all attributes, and then unified mapping binding route. In Ant Design Pro, the routing system of the page layout is closely related to the navigation menu. Routing uses the routing policy of centralized management of configuration data to map the configuration data of the navigation menu one by one. In router.js, the route uses the require method to request each page Java Script resource corresponding to path and mount it in Component. Each page Java Script resource corresponds to Path. This way, when a particular routing address is requested, Java Script resources can be loaded to render the desired page.

The Authorized file is introduced to identify and manage the access permission of each route. If the user permission is different from the access permission of the current requested route, the redirect path is configured in the navigation menu. The React. Create Element method creates the DOM of the page. If the browser sends a request for the router address and the system provides the corresponding Java Script resource, the 404 page will be displayed. Rewrite menu Data Render in Ant Design Pro's source code file Basic Layout.js, and then convert the Data source to the Data input by the user, and then import the generated JSON configuration Data into the automatically generated navigation menu.

Finally, the navigation configuration data is mounted to the component corresponding to each route. Similarly, the menu data in the scaffold source code is rewritten, and the rewritten input parameter data source is taken as the data of user input Settings. During this period, the authorizations of all menu items need to be checked by means of the Authorization. check method.

According to the above design method, dynamic configuration interface of navigation menu can be obtained. The user has the right to edit the configuration and modify the page name, navigation menu icon, routing address, etc. After setting the navigation menu, the configuration data of the navigation menu will be automatically generated and stored in the local browser.

The navigation menu generated in this paper is shown in Figure 4.

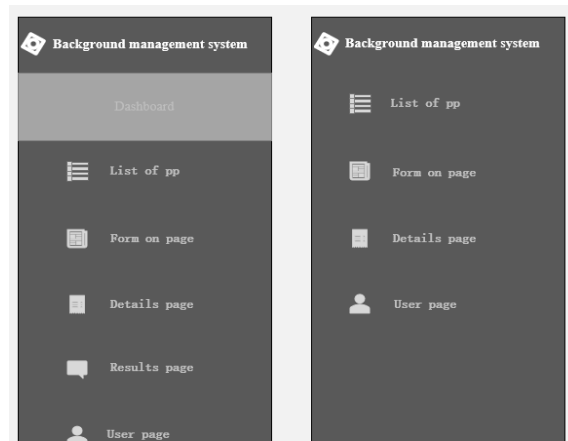


Figure 4: Navigation menu

4.2 Implementation of Component Library Function Modules

The component library module is implemented as follows: First, encapsulates the main ECharts chart components required by the background management system including interactive functions, including bar charts, line charts, pie charts and other types; Second, with the aid of Ant Design, the logic function of the encapsulated data Table components, such as List, Table, From, etc. Specifically, first of all, according to the functional requirements, the Java Script code is segmented to encapsulate the style and logic into independent front-end components that can interact, so as to form a component library with interactive functions. When users browse components in the component library panel and click to select, the system automatically identifies and loads components to the current page. Secondly, the request data interface and other method functions are encapsulated to improve the logic of the code, enhance the method call between various functions, and further reduce the code repeatability. Finally, the echarts diagram component is constructed by using Java Script library resource files in Echarts-for-React. Echarts-for-react is the react component encapsulation of Echarts. When echarts-for-React encapsulates the native API, React will retain the features of the API, greatly reducing design and development costs and simplifying code. In this case, you can use React to call the Echarts interface to render the Echarts chart directly, and then import xangguan parameters or data to obtain the required chart. Based on the appeal process, the final component browsing operation interface is obtained, as shown in Figure 5.

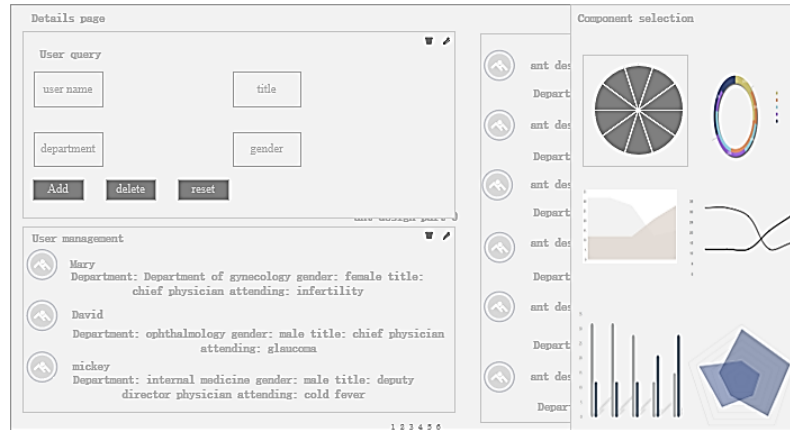


Figure 5: Component library function module operation browsing interface

As you can see, the image rendering of each component is displayed on a panel in the form of an interactive drawer that can be scaled on the right side of the page for easy use. When the user clicks a selected component, the component can be added to the current page by clicking the addITEM method of the event binding.

4.3 Component Editing Function Module Implementation

Component editing function module implements visual dynamic editing Settings for main attribute parameters of components, including style attribute parameter setting, binding back-end interface, interface parameter setting, component display data setting and other functions. In the component library module has been implemented to parameterize the properties of components, and extract their properties as component API, used to edit the configuration of components. After configuring parameters using the component editing module, you can dynamically modify and edit components and automatically generate component configuration data for saving.

The Settings of component configuration data are encapsulated as common functional methods for reuse. In order to facilitate visual configuration and use and preview editing effects, edit panel components are prepared, which are displayed in the left side of the interface in the form of drawer interaction for expansion. Interface data configuration panel to set the interface parameters, first of all to edit the interface URL parameters; Secondly, select get, POST, PUT and other options in the drop-down box for the request mode. Then in the parameter list for the input request parameters key-value configuration, and set the data type verification, the parameters can be num type and string type conversion, to meet the requirements of common interface request parameters. Finally, you can specify the data source key for the result returned by the request. This module binds to the

request method encapsulated in the previous section and is configured by the user to request the interface directly and return render data. It can efficiently and intuitively bind data interfaces, which is similar to a simple interface testing tool, simplifying the traditional complex process from interface testing to front-end data rendering. Component style property parameter Settings, including component title, location size parameters; For Echarts components, more style configuration data can be visually edited, allowing users to customize chart component styles with simple actions.

The component Style edit panel also encapsulates a color picker that allows you to set the background color of the component. When you edit a component, you can preview the effect of modifying parameters on the component in real time. The component display data configuration module mainly performs a one-to-one key-value binding configuration on the display data obtained from back-end interface data fields or operation request parameters on the component. The specific configuration interface is shown in Figure 6.

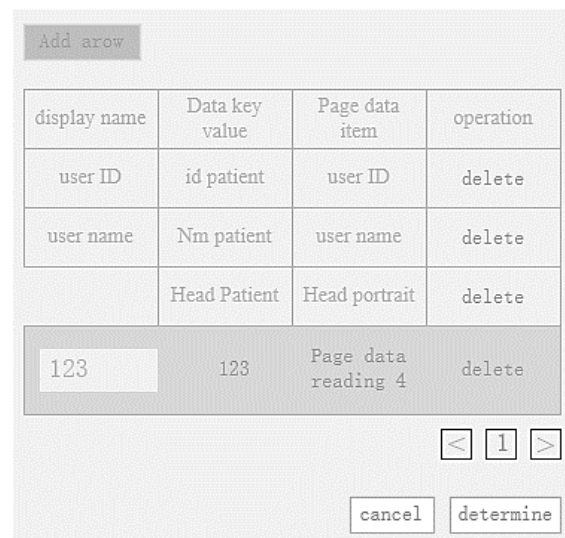


Figure 6: Component shows the data configuration interface

4.4 Configuration Data Operation Module Implementation

The implementation of configuration data operation module includes browser local real-time storage and reading, database persistent storage and reading these two modules, for one, users can according to needs, the system configuration data, through the back-end interface into the database, complete data persistent storage and request. The system uses the user's manual operation to upload and read the formal data file of the database. When uploading data or closing the browser window, a window will pop up to remind the user to save and confirm the operation, as shown in Figure 7.

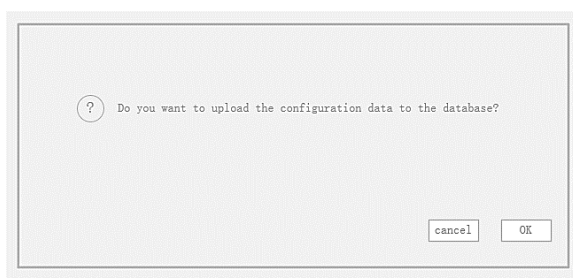


Figure 7: Data upload Confirm popup

Then, the fetch function encapsulated above is called by the save method to request the back-end interface, and the configuration data will be processed by the back-end application and saved to the database to complete the persistent storage of data. If the chrome developer tool displays the upload interface request status code 200, configuration data is successfully uploaded. Secondly, the design of the data stored in the database to read the function, the user can choose whether to load the database saved data files.

The popup window is used to remind the user, and when the user agrees to load, the configuration data operation module reads the data of the latest version saved in the database and renders it in the interface, as shown in Figure 8.

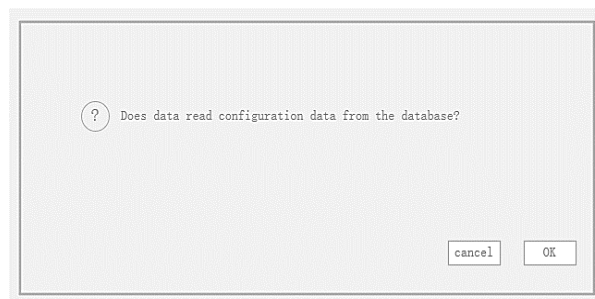


Figure 8: Data read request popup

The same as the configuration data upload step, the encapsulated FETCH interface requests the back-end reading interface, and then the @Connect decorator realizes the receiving and binding of the Redux store corresponding to the storage configuration data, that is, the list data in datalist. Finally, the list data is parsed with json.parse method, so as to complete the reading of configuration data in the database. Similarly, if the data read interface request status code is 200 in the Chrome developer tool, the data is successfully read.

5. System Test

In order to test the performance of the system, three tests were carried out: first, 30 tests were carried out in different test environments to clarify its rendering capability. During this test, the test environment is the mainstream browser at home and abroad, as shown in Table 2.

Table 2. Introduction of different test environments

browser	IE	Chrome	Firefox	safari	QQ browser	360 browser
Browser Version	11.0	87.0	86.0	11.0	9.4	13.1
operating system	Windows	Windows	Windows	macOS Siema	Windows	Windows
kernel	Trident	Blink	Gecko	Wehkit	Chromium +Trident	Chromium +Trident
System version	Windows 10	Windows 10	Windows 10	10.12.6	Windows 10	Windows 10

The test results in each test environment were obtained through this test, and the mean value was processed.

The final results are shown in Table 3. The test found that the page running latency was less than 2s.

Table 3. Rendering results under different test environments

browser	IE	Chrome	Firefox	safari	QQ browser	360 browser
operating system	Windows	Windows	Windows	macOS Siema	Windows	Windows
Render time /ms	1838	1562	936	643	1725	1505

The test found that there was a delay in page running, but the time was less than 2s, which was mainly caused by the loading calculation of JavaScript resources. Therefore, there is still room for optimization in subsequent studies, but in general, the rendering time of the system in different simultaneous environments is relatively good, which can meet the needs of practical applications. Secondly, the unit test is carried out, through which the system module can be tested effectively and the problem can be located accurately. Specifically, unit test is carried out with the help of Jest testing framework, that is, the unit test case code created for the permission function and navigation routing function, when the test is successful, the routing address and navigation data generated by it can be successfully printed in the terminal. Thirdly, cross-device response test. In order to test the cross-device response of the system, Huawei Honor 60 was selected with a 6.67 inch screen and 2400*1080 pixels resolution. The device was used to preview the system page and found that the system had a good sense of accessories. In summary, the above tests confirm the feasibility of the system design, which can be applied reasonably in the subsequent practice.

6. Conclusions

In general, this paper first made a preliminary understanding of React componentization technology, which laid a foundation for the subsequent system design. Secondly, after theoretical knowledge, the functional modules of the componentized background management visual page construction system are designed in detail, including page editing module, component library module, component editing module, etc. Finally, according to the design process, visualization of componentization backstage management page to achieve the function of building the system, and the robustness of the system by means of empirical, the inspection of the empirical results are three: first, under different test conditions, the page operation delay time are less than 2 s, have a better rendering results; Second, the generated routing address and navigation data can be successfully printed in the terminal.

Third, whether huawei Honor 60 or tablet computer, can fully display a good system page, with a good sense of accessories. Therefore, the robustness of the system is confirmed and can be reasonably extended in practice.

References

- [1] Guo R, Zhao YS. "Overview of Web Front-end Framework Technology," *Journal of Beijing Polytechnic Institute*, 2019, 20(03):24-27.
- [2] Wang M, Tian Y, Li NN. "Modular WEB front-end architecture design and implementation of," *Computer Knowledge and Technology*, 2018, 14(30):77-79.
- [3] Du YM, Huang XF. "Model and practice of front-end separation development for enterprise web applications," *Journal of Southwest University of Science and Technology*, 2018, 33(02):83-87.
- [4] Wang Yi, Zhao JX, Feng HY, Xue K. "Modular data show software design and implementation of the prison," *Computer Measurement and Control*, 2017, 25(11):103-106.
- [5] Xue YY, Liu HT, Zhao G. "Design method of information development framework based on Intelligent components," *Information Technology and Informatization*, 2019, (08):18-20.
- [6] Yin T, Cui JD. "Research on componentization of development based on Android software," *Computer Applications and Software*, 2019, 36(09):16-20.
- [7] Li YM, Kang JR, Li X. "Analysis of iOS componentized Routing Design," *New Industrialization*, 2020, 10(8):8-9 + 13.
- [8] Yang RB, Du B, Yang C, Jiao PF, Zhou G. "General command system framework design based on component technology," *Journal of Information Technology and Network Security*, 2020, 33 (09) 6:79-82.
- [9] Fang S. "Based on the "Vue. Js" front-end framework technology research," *Computer Knowledge and Technology*, 2021, (19):59-60 + 64.
- [10] Xiong JL. "Research and application of Web Front-end componentization based on React," *Chongqing University of Posts and Telecommunications*, 2018.